Privacy-Preserving and Fraud-Resistant Targeted Advertising for Mobile Devices
Cardiff University

Stylianos S. Mamais

21 February 2019
Abstract

Online Behavioural Advertising (OBA) enables Ad-Networks to capitalize on the popularity of digital Publishers in order to target users with context-aware promotional materials from Advertisers. OBA has been shown to be very effective at engaging consumers but at the same time presents severe privacy and security threats for both users and Advertisers. Users view OBA as intrusive and are therefore reluctant to share their private data with Ad-Networks. In many cases this results in the adoption of anti-tracking tools and ad-blockers which reduces the system's performance. Advertisers on their part are susceptible to financial fraud due to Ad-Reports that do not correspond to real consumer activity. Consequently, user privacy is further violated as Ad-Networks are provoked into collecting even more data in order to detect fictitious Ad-Reports.

Researchers have mostly approached user privacy and fraud prevention as separate issues while ignoring how potential solutions to address one problem will effect the other. As a result, previously proposed privacy-preserving advertising systems are susceptible to fraud or fail to offer fine-grain targeting which makes them undesirable by Advertisers while systems that focus on fraud prevention, require the collection of private data which renders them as a threat for users. The aim of our research is to offer a comprehensive solution which addresses both problems without resulting in a conflict of interest between Advertisers and users. Our work specifically focuses on the preservation of privacy for mobile device users who represent the majority of consumers that are targeted by OBA. To accomplish the set goal, we contribute ADS+R (Advert Distribution System with Reporting) which is an innovative advertising system that supports the delivery of personalized adverts as well as the submission of verifiable Ad-Reports on mobile devices while still maintaining user privacy. Our approach adopts a decentralized architecture which connects mobile users and Advertisers over a hybrid opportunistic network without the need for an Ad-Network to operate as administrative authority. User privacy is preserved through the
use of peer-to-peer connections (serving as proxy connections), Anonymous-
download technologies and cryptography, while Advertiser fraud is prevented
by means of a novel mechanism which we termed Behavioural Verification.
Behavioural Verification combines client-side processing with a blockchain-
inspired construction which enables Advertisers to certify the integrity of
Ad-Reports without exposing the identity of the submitting mobile users.
In comparison to previously proposed systems, ADS+R provides both (1)
user privacy and (2) advert fraud prevention while allowing for (3) a tunable
trade-off between resource consumption and security, and (4) the statistical
analysis and data mining of consumer behaviours.
Dedication

To my beloved mother who supported and funded my academic career.
Declaration

I declare that the content of this thesis is the product of my own research.
Acknowledgements

I would like to thank my supervisor, Dr. George Theodorakopoulos who was always available to point me in the right direction. I am also grateful to my co-supervisor, Dr. Stuart Allen for giving me very useful insights during my research.
## Contents

1 Introduction .................................................. 14
   1.1 Problem Statement ...................................... 17
   1.2 Research Aim ............................................ 17
   1.3 Proposed System Overview .............................. 17
   1.4 Contributions ........................................... 19
   1.5 Publications ............................................ 20
   1.6 Thesis Structure ......................................... 21

2 Background .................................................. 22
   2.1 OBA: Online Behavioural Advertising ................. 22
   2.2 Consumer Tracking ...................................... 24
   2.3 Privacy Concerns and Countermeasures ................. 28
   2.4 Advertising Fraud ...................................... 29
   2.5 Advertising Fraud Detection ............................ 31
   2.6 OBA Issues and Solution Approaches .................. 32

3 Related Work .................................................. 34
   3.1 Advertising Privacy .................................... 34
      3.1.1 Literature Review .................................. 36
      3.1.2 Limitations of Advertising Privacy Systems .... 42
   3.2 Fraud Prevention ........................................ 44
      3.2.1 Literature Review .................................. 45
      3.2.2 Limitations of Fraud Prevention Systems ......... 47
   3.3 Related Work Summary .................................. 48

4 ADS: Advert Distribution System ............................ 50
   4.1 System Specifications ................................... 51
      4.1.1 Stakeholders ........................................ 51
      4.1.2 Trust Model ......................................... 52
6.2.5 Behavioural Verification ........................................ 107
6.2.6 Statistical Analysis of Consumer Behavioural Patterns 113
6.3 Protocol ................................................................. 114
6.4 Evaluation ............................................................... 118
  6.4.1 Reporting Effectiveness ......................................... 118
  6.4.2 Reporting Fraud Prevention .................................... 118
  6.4.3 Reporting Integrity ............................................... 122
  6.4.4 User Privacy .................................................... 124
6.5 ADS+R: Advert Fraud Prevention Summary ................. 124

7 Conclusion ........................................................................ 125
  7.1 System Limitations .................................................. 127
  7.2 Practical Implementation ......................................... 128
  7.3 Future Work ......................................................... 130
  7.4 Final Remarks ...................................................... 131
List of Figures

2.1 OBA (Online Behavioural Advertising) model architecture. 25

4.1 ADS (Advert Distribution System) architecture. 55
4.2 Visual representation of the handshake protocol which is used by ADS for user authentication. 59
4.3 Visual representation of the ARM (Ad-Request Message) forwarding sub-protocol. 62
4.4 Example of the ARM (Ad-Request Message) forwarding sub-protocol. 63
4.5 Advert Collection sub-protocol. 67

5.1 Demographic attributes which was supported by D-PC (Demographic Profile Comparison). 77
5.2 Example of the ranking process which is performed by F-PC (Fragmented Profile Comparison). 81
5.3 Success rate contrast of F-PC (Fragmented Profile Comparison) and random selection with randomly generated profiles. 86
5.4 Success rate contrast of F-PC (Fragmented Profile Comparison) and random selection with profiles generated from a dataset. 87
5.5 Success rate contrast of F-PC (Fragmented Profile Comparison) and random selection with profiles generated from Pareto principle. 88
5.6 Delivery efficiency of F-PC (Fragmented Profile Comparison) in comparison to a random selection. 90

6.1 Advert Distribution System with Reporting (ADS+R) architecture. 98
6.2 Supported types of Ad-Reports and their contented elements. 104
6.3 Structural information components of ADS+R. 105
LIST OF FIGURES  

6.4  Visual representation of the *CS-Board* (Service Confirmation Board). .................................................. 107
6.5  Example of Behavioural Verification through advert association.109
6.6  Example of Behavioural Verification through the use of *CB* (Checkpoint Block). ................................. 110
6.7  Example of Behavioural Verification through the use of *AB* (Affiliation Block). ............................ 112
6.8  Example of Behavioural Verification through the combination of all available methods. ......................... 113
6.9  Report Form Collection sub-protocol. ............................ 115
6.10 Ad-Report Submission sub-protocol. ............................ 117
List of Tables

3.1 Evaluation table of advertising privacy preserving systems. . 44
4.1 Table of trust relations between $ADS$ stakeholders. . . . . . 53
5.1 Evaluation table of profile comparison methods. . . . . . . 94
6.1 Table of trust relations between $ADS+R$ stakeholders. . . . 100
## Abbreviation Index

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>Affiliation Block</td>
</tr>
<tr>
<td>ADK&lt;sup&gt;sig&lt;/sup&gt;</td>
<td>Ad-Dealer Signing Key</td>
</tr>
<tr>
<td>ADK&lt;sup&gt;ver&lt;/sup&gt;</td>
<td>Ad-Dealer Verification Key</td>
</tr>
<tr>
<td>ADLT</td>
<td>Ad-Dealer Location Tag</td>
</tr>
<tr>
<td>ADS</td>
<td>Advert Distribution System</td>
</tr>
<tr>
<td>ADS+R</td>
<td>Advert Distribution System with Reporting</td>
</tr>
<tr>
<td>Aid</td>
<td>Ad-Dealer Identity</td>
</tr>
<tr>
<td>AIP&lt;sub&gt;u&lt;/sub&gt;</td>
<td>Advertising Interest Profile of user u</td>
</tr>
<tr>
<td>ARC</td>
<td>Ad-Report Chain</td>
</tr>
<tr>
<td>ARC-ID</td>
<td>Ad-Report Chain Identity</td>
</tr>
<tr>
<td>ARM</td>
<td>Advert Request Message</td>
</tr>
<tr>
<td>AW&lt;sub&gt;AD&lt;/sub&gt;</td>
<td>Average Wait of user u</td>
</tr>
<tr>
<td>A-Token</td>
<td>Action Token</td>
</tr>
<tr>
<td>BroK&lt;sup&gt;Privi&lt;/sup&gt;</td>
<td>Broker Private Key</td>
</tr>
<tr>
<td>BroK&lt;sup&gt;Pub&lt;/sup&gt;</td>
<td>Broker Public Key</td>
</tr>
<tr>
<td>BundleID</td>
<td>Bundle Identity</td>
</tr>
<tr>
<td>CB</td>
<td>Checkpoint Block</td>
</tr>
<tr>
<td>ConPW&lt;sub&gt;u1&lt;/sub&gt;&lt;sup&gt;2&lt;/sup&gt;</td>
<td>Contact Authentication Passwords of users u1 and u2</td>
</tr>
<tr>
<td>CS&lt;sub&gt;i&lt;/sub&gt;</td>
<td>Candidate Selection</td>
</tr>
<tr>
<td>CS-PC</td>
<td>Collaborative-Selective Profile Comparison</td>
</tr>
<tr>
<td>CT&lt;sub&gt;u&lt;/sub&gt;</td>
<td>Collection Time of user u</td>
</tr>
<tr>
<td>C-Token</td>
<td>Click Token</td>
</tr>
<tr>
<td>DelK&lt;sub&gt;i&lt;/sub&gt;</td>
<td>Delivery Key of user u</td>
</tr>
<tr>
<td>DM</td>
<td>Delivery Message</td>
</tr>
<tr>
<td>DT&lt;sub&gt;u&lt;/sub&gt;</td>
<td>Delivery Time of user u</td>
</tr>
<tr>
<td>D-PC</td>
<td>Demographic Profile Comparison</td>
</tr>
<tr>
<td>EToC&lt;sub&gt;u&lt;/sub&gt;</td>
<td>Estimated Time of Collection of user u</td>
</tr>
<tr>
<td>EToD&lt;sub&gt;u&lt;/sub&gt;</td>
<td>Estimated Time of Delivery of user u</td>
</tr>
<tr>
<td>F-PC</td>
<td>Fragmented Profile Comparison</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$I_{id}$</td>
<td>Interest Identifier</td>
</tr>
<tr>
<td>$ID_u$</td>
<td>Identity Name of user $u$</td>
</tr>
<tr>
<td>$IH$</td>
<td>Integrity Hash</td>
</tr>
<tr>
<td>$K^i$</td>
<td>Key $i$</td>
</tr>
<tr>
<td>$MPW_u$</td>
<td>Master Password of user $u$</td>
</tr>
<tr>
<td>$OrderID$</td>
<td>Order Identity</td>
</tr>
<tr>
<td>$P_{id}$</td>
<td>Publisher Identity</td>
</tr>
<tr>
<td>$RepK_u^{\text{sig}}$</td>
<td>Report Signing Key of user $u$</td>
</tr>
<tr>
<td>$RepK_u^{\text{ver}}$</td>
<td>Report Verification Key of user $u$</td>
</tr>
<tr>
<td>$RF$</td>
<td>Report Form</td>
</tr>
<tr>
<td>$RoA$</td>
<td>Report of Action</td>
</tr>
<tr>
<td>$RoC$</td>
<td>Report of Click</td>
</tr>
<tr>
<td>$RoV$</td>
<td>Report of View</td>
</tr>
<tr>
<td>$RT_{user}$</td>
<td>Run Time of user $u$</td>
</tr>
<tr>
<td>$SC-Board$</td>
<td>Service Confirmation Board</td>
</tr>
<tr>
<td>$SN_i$</td>
<td>Sequence Number</td>
</tr>
<tr>
<td>$SysDK$</td>
<td>System Decryption Key</td>
</tr>
<tr>
<td>$SysEK$</td>
<td>System Encryption Key</td>
</tr>
<tr>
<td>$S-PC$</td>
<td>Selective Profile Comparison</td>
</tr>
<tr>
<td>$ToK_A^{\text{sig}}$</td>
<td>Token Signing Key of Ad-Dealer $A$</td>
</tr>
<tr>
<td>$ToK_A^{\text{ver}}$</td>
<td>Token Verification key of Ad-Dealer $A$</td>
</tr>
<tr>
<td>$T\text{-Token}$</td>
<td>Time Token</td>
</tr>
<tr>
<td>$VC-i$</td>
<td>Verification Check of Block $i$</td>
</tr>
<tr>
<td>$VC-I$</td>
<td>Verification Check of Issuing</td>
</tr>
<tr>
<td>$VC-S$</td>
<td>Verification Check of Submission</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

The digital advertising market has exhibited a steady expansion over the last two decades [76]. The first digital advert was published by HotWire.com on October 27, 1994 and shortly after digital advertising started to become adopted across the internet [84]. Much like their printed counterparts, the first digital adverts were simple static banners that were presented at website viewers. However, it soon became evident that digital platforms offered the potential for more sophisticated advertising methods that were not previously possible. Unlike traditional media such as televising, radio, magazines and billboards which are directed at large audiences, digital media are accessed by individual consumers, thus allowing for the illustration of personalized content. To fully exploit the potential of digital media, marketers began to develop elaborate methods for segmenting audiences to evermore refined consumer groups [62]. Eventually, marketers were able to target distinctive users based on their personal demographic traits and online habits. This was made possible through an approach known as Online Behavioral Advertising (OBA). The United States Federal Trade Commission (FTC) defines OBA as ‘the practice of tracking an individual’s online activities in order to deliver advertising that is tailored to the individual’s interests[61]. In practical terms, OBA is an advanced marketing technique for determining a user’s consumer needs for the purpose of matching them with specific adverts.

Compared to non-targeted forms of advertising, OBA has been shown to be more adept at engaging consumers [21, 153, 16, 26, 25, 55]. For Advertisers, greater user engagement practically signifies increased sales while at the same time allowing them to build brand recognition and loyal customer following [74]. An assessment by Google estimates that in 2017, their adver-
CHAPTER 1. INTRODUCTION

Advertising services were responsible for generating 283 billion US dollars for more than 1.5 million businesses in the US [64]. For the second quarter of 2017, Facebook reported an advertising revenue of 9.16 billion US dollars while additional research suggests that 26% of Facebook users who clicked on an advert reported making a purchase [10, 129]. Currently, Google and Facebook dominate the digital advertising industry with a combined US market share of 63% [45].

Advertisers were quick to recognize the profitability potential of OBA which provoked an increased interest towards digital Publishers [13]. Estimates for 2018 show that 61% of all adverts in the United Kingdom are published on digital media while digital advert spending in the US is calculated at 93.7 billion US Dollars which amounts to 43.6% of the country’s total marketing budget with estimates for a steady increase to 51.3% by 2021 [40, 12]. On a global scale, digital advert spending is valued at 273.3 billion US Dollars with analysts foreseeing a gradual rise by the year 2022 to 427.2 billion US Dollars that will account for 53.9% of all advert spending [47].

Publishers on their part, view OBA as a viable business model. By capitalizing on the popularity of their platforms, Publishers are able to generate sufficient revenue through adverts, thus being able to offer their content for no added cost. In that regard, digital advertising also provides an indirect benefit for users who gain access to the free Publisher services such as informative websites, social networks, hosting platforms, email, instant messaging, cloud storage and freeware (free software). Indicative of how much users gain from advertising is the fact that 95% of apps for Android and 88% for iOS are offered to users for free [11, 131]. Arguably, it would not have been possible for developers to offer their mobile apps for free if not for the prospect of advertising revenue.

From a business point of view, it can be stated that OBA is mutually beneficial for all participating stakeholders, including users. In a way, OBE has helped shape digital media and at the same time revolutionized the marketing industry [53]. Nevertheless, that is not to say that OBA does not also come with important concerns, limitations and drawbacks. It has long been argued that OBA presents a significant threat for user privacy [109, 128, 85, 116, 33]. Directly effected are billions of digital users with estimates indicating that 80% of the population in North America and Western Europe is accessing the internet on a daily basis [49]. This accounts for nearly 280 million users in the US alone with figures expected to rise by 2019 to more than 3.84 billion people worldwide [48, 44]. Although most users hold misconceptions about OBA and are not fully aware of the extent
of the threat to their privacy, the vast majority denounce the way that their information is being exploited when informed about the invasive nature of OBA [105, 104, 139]. Recent surveys confirm the negative sentiment towards OBA with 71% of US users reporting a noticeable upsurge in the intrusiveness of targeted adverts in the last three years while in the UK only 6% of users are expressing a liking in re-targeted adverts (adverts for products they have previously browsed) [52, 50]. Regardless of their privacy concerns, consumers still view tailored adverts as useful but the lack of available alternatives to OBA has prompted many of them to reject advertising altogether [141]. As a result, consumers are turning to the adoption of add-on software such as anonymizers, cookie managers and ad-blockers. Estimates for 2018 show that 12.2 million users in the UK and as many as 21.4 million in Germany are running ad-blocking software [46]. The wide proliferation of ad-blockers has caused notable concerns to the marketing industry with many researchers proclaiming that the viability of internet advertising is under threat [126, 127, 123, 97]. As a response, some Publishers have begun to limit access to their content for users who are running ad-blockers [110].

From the perspective of Advertisers, even more concerning than the losses due to adware and ad-blockers is the susceptibility of the currently implemented OBA model to fictitious advert views and clicks. Attackers exploit the system’s vulnerability in order to defraud Advertisers either for personal financial gain or with the intent to engage in corporate espionage against competitors by depleting their advertising budget [59]. To conduct these attacks, the perpetrators can either employ the services of human operators to click on adverts (known as click-farms) or use automated malware programs such as auto-clickers and clickbots [34]. Clickbots, in particular, have been widely used to conduct large-scale fraud to a great effect [35]. Attributing to the widespread use of clickbots is their effectiveness, accessibility, low cost and ability to avoid detection [149, 106]. Researchers indicate that the majority of popular mobile Ad-Networks are liable to attacks from clickbots [27]. Estimates by eMarketer show that up to 10% of the marketing budget of UK businesses is vulnerable to advert fraud while other reports suggest that Advertisers are likely to lose up to 51 million US dollars per day in 2018, totaling 19 billion US dollars over the entire year with figures expected to rise up to 44 billion by 2022 [51, 120]. To combat advert fraud, Ad-Networks attempt to detect invalid traffic through the enforcement of policy-based filtering mechanisms but this approach has not been entirely effective [107]. As a response, researchers have proposed the adoption of more sophisticated filtering methods [83, 145, 81, 157]. For traffic filtering
to be feasible however, Ad-Networks will be inclined to collect even more data which will further infringe user privacy. Although the technologies that are used to filter advertising reports are proprietary and therefore not accessible to the public, Advertisers have been critical of Ad-Networks for not pursuing fraud detection aggressively enough [79]. To support this claim, academics have pointed out that Ad-Networks may be biased towards the matter as both themselves as well as Publishers directly benefit from invalid traffic [88, 151]. A more suitable approach for the marketing industry would therefore be to assign the auditing of traffic to a third party [151]. For such an approach to be adopted, the Advertisers and Ad-Networks would need to cooperate and agree on the establishment of an independent authority which would be entrusted to manage the submitted Ad-Reports. However, it would be highly unlike for Ad-Networks to agree on such a setup as it would require the third party to be granted direct access to the Ad-Network’s system which would be considered a serious security risk.

1.1 Problem Statement

Previous attempts to address user privacy and fraud prevention as separate issues have been able to partially resolve one of the two problems, but only at the expense of the other. More specifically, privacy-preserving advertising systems are susceptible to fraud, or fail to offer fine-grain targeting (present tailor made adverts which match specific user interests), making them undesirable by Advertisers while systems that focus on fraud prevention require the collection of private data which renders them as a threat for users. To the best of our knowledge there has never been a comprehensive solution that is mutually beneficial for both users and Advertisers.

1.2 Research Aim

The aim of our work can be summarized as follows: We aim to offer an alternative implementation of OBA (Online Behavioural Advertising) for mobile devices, which provides both user privacy and Advertiser protection against advertising fraud while retaining fine-grain targeting capability.

1.3 Proposed System Overview

In this thesis we introduce ADS+R (Advert Distribution System with Reporting) as an innovative advertising system which supports the delivery of
personalized adverts as well as the submission of verifiable Ad-Reports on mobile devices while still maintaining user privacy. Our approach adopts a decentralized architecture which connects mobile users and Advertisers over a hybrid opportunistic network without the need for an Ad-Network to operate as administrative authority. The system takes advantage of client-side processing to allow users to compose their own Ad-Requests and Ad-Reports. The Ad-Requests are generated based on a locally stored user interest profile which can support fine-grain advert targeting while the Ad-Reports do not contain any information which could compromise the user’s identity. The hybrid opportunistic network is then used to deliver both Ad-Requests and Ad-Reports to the Advertisers for processing. Ad-Requests are answered with matching adverts while Ad-Reports are independently verified and awards are issued to the concerned Publishers.

User privacy against Advertisers is preserved through the use of peer-to-peer connections which serve as partially trusted proxies and anonymous-download technologies that allow mobile devices to transfer data without compromising the user’s identity. To further enhance privacy and security, cryptography is applied on both the Ad-Requests and Ad-Reports in order to prevent intermediate network nodes and eavesdroppers from obtaining private user information or sabotaging the system by injecting fake adverts and other malicious content.

ADS+R also preserves bandwidth and memory by enabling multiple users to collectively have access to the same encrypted adverts while still maintaining their privacy. To allow users to identify adverts that may be of shared interest, the system incorporates four profile comparison algorithms Demographic Profile Comparison (D-PC), Fragmented Profile Comparison (F-PC), Selective Profile Comparison (S-PC) and Collaborative-Selective Profile Comparison (CS-PC). The aforementioned profile comparison algorithms allow for a trade off between resource conservation and privacy which is tunable based on the individual user’s preferences.

Advertiser fraud is prevented by means of a novel mechanism which we termed Behavioural Verification. Behavioural Verification combines client-side processing with a blockchain-inspired construction in order to classify users as honest or dishonest. What constitutes a user’s honesty for our system is the manner in which they access adverts on their mobile device. Dishonest users submit multiple reports over a short period of time while honest users behave as consumers who view adverts at a balanced pace while engaging in typical social activities such as making online purchases, moving through space and interacting with other users. We argue that it is hard for dishonest users to fake honest behaviour and we exploit social behavioural
patterns to identify fraudulent Ad-Reports without compromising the identity of the submitting users. A supplementary feature of our approach is that it also enables us to perform anonymous statistical analysis of consumer behavioral patterns. This includes the ability to identify correlations between advertising interests (e.g., users with interest in product A are also interested in product B), visited location (e.g., users that visit location A are interested in product B) and social affiliations (e.g., users with interests in product A are closely affiliated with users with interest in product B).

ADS+R entirely disrupts the operation of the currently iterated OBA system as it makes Ad-Networks obsolete and outsources their functionality to the users, Advertisers and Publishers. More specifically, ADS+R exploits the processing power of mobile devices in order to allow users to locally determine their advertising needs and collect the appropriate targeted adverts from the Advertisers. In a similar fashion, the users also compose and submit their own Ad-Reports which are independently verified by the concerned Advertisers and can then be shared with the Publishers. For both operations, users, Advertisers and Publishers establish a direct line communication over a decentralized network without the need for an Ad-Network to operate as administrative authority. The decentralized architecture of ADS+R is easy to establish and fully supported by currently available networking technologies such as 5G. Client-side processing is also feasible by currently available smart-phone devices and arguably more effective than cloud-based processing as it offers greater targeting accuracy when considering the fact that local services have direct access to user data while cloud-hosted services need to rely on tracking protocols and voluntarily shared data. Lastly, ADS+R also offers greater flexibility for Advertisers as it supports the fusion of multiple targeting algorithms. In turn, this will allow Advertisers to tailor the system’s functionality to their own needs and usher the development of new targeting approaches.

1.4 Contributions

We claim the following contributions:

- **Privacy-preserving distribution of targeted adverts**: ADS which was published in [99] and is presented in Chapter 4 is an advertising system which supports fine-grain targeting while providing complete mobile user privacy against all other stakeholders (including other users) and security against sabotage from attackers.
• **Resource conservation within opportunistic networks:** The four profile comparison algorithms (*D-PC, F-PC, S-PC* and *CS-PC*) which are presented in Chapter 5 can be used to conserve bandwidth and memory within an opportunistic network by identifying similarities between the advertising interests of system users (network nodes) and allowing them to share access to the same adverts without compromising their privacy.

• **Privacy-preserving detection of advert fraud:** *ADS+R* which was published in [98] and is presented in Chapter 6 is an extension of *ADS* (see Chapter 4) which enables *Advertisers* to independently verify the validity of submitted *Ad-Reports* without requiring any trust towards a third party (an *Ad-Network*) while maintaining the privacy of the submitting mobile users.

  – **Anonymous statistical analysis of consumer behaviours:** One of the features of *ADS+R* which is presented in Section 6.2.6 is the anonymous statistical analysis of consumer behavioral patterns. This includes the ability to identify correlations between advertising interests, visited location and social affiliations.

### 1.5 Publications

The research presented in this thesis has contributed to the following publications:

• "Private and secure distribution of targeted advertisements to mobile phones." Future Internet 9.2 (2017): 16.


The first publication titled "Private and secure distribution of targeted advertisements to mobile phones." consists the entirety of the research which is presented in Chapter 4 and an initial version of the research work that can be found in Section 5.1.2 of Chapter 5. The second publication titled "Behavioural Verification: Preventing Report Fraud in Decentralized Advert Distribution Systems." consists the entirety of the research which is presented in Chapter 6.
1.6 Thesis Structure

The structure of this thesis is as follows. Chapter 2 offers a preliminary knowledge of the OBA model and a technical overview of the particular threats which are faced by the marketing industry. Chapter 3 presents a detailed analysis and scrutiny of the related work in regards to preservation of user privacy as well as the prevention of advert fraud. In Chapter 4 we present ADS (Advert Distribution System), a novel approach for distributing personalized adverts over a social network of mobile users while preserving user privacy. In Chapter 5 we present four profile comparison algorithms (D-PC, F-PC, S-PC and CS-PC) which can be used as an add-on to ADS. Our profile comparison algorithms aim to conserve memory and bandwidth within the opportunistic network that ADS operates on by identifying users (network nodes) with the same consumer interests and allowing them to share access to the same content without compromising their privacy. In Chapter 6 we present ADS+R (Advert Distribution System with Reporting) which is an extension of ADS. ADS+R utilizes the same infrastructure as ADS but also introduces the concept of Behavioural Verification. Behavioural Verification is a novel approach for preventing advert fraud while still maintaining user privacy and also allows for the statistical analysis of consumer behavioural patterns. Lastly, in Chapter 7 we summarize our work, reflect on the innovations and limitations of our system and discuss our perspectives for future research.
Chapter 2

Background

In this chapter we offer a brief insight on advertising technologies and the manner in which they are applied. In particular, in Section 2.1 we introduce the stakeholders of the OBA (Online Behavioural Advertising) ecosystem and detail their operation. In Section 2.2 we analyze the various user tracking methods which are available and then proceed to explain how they violate user privacy in Section 2.3. In Section 2.4 we analyze the most prominent forms of advertising fraud and lastly in Section 2.5 we provide an overview of the methods which are currently being used to combat them.

2.1 OBA: Online Behavioural Advertising

*Online Behavioural Advertising (OBA)* is an advanced marketing method for targeting digital users with context-aware adverts. The OBA ecosystem consists of four parties as shown in Figure 2.1. *Advertisers* are representatives of businesses who wish to promote products and services through advertising campaigns. Users represent potential consumers who may be interested in a particular product or service which is offered by Advertisers. *Publishers* are digital platforms such as websites, software applications or other services which attract user traffic. Lastly, *Ad-Networks* are companies such as Google Ads and Yahoo! AdNet which operate as a middleman between Advertisers, Publishers and users.

To take part in the OBA system, Advertisers create promotional materials (digital adverts) and supply them to an Ad-Network. Publishers on their part, reserve within their user interfaces (whether websites or software) certain visual areas which can be used by an Ad-Network for the illustration of adverts. Known as Ad-Boxes, these visual areas are controlled by the
CHAPTER 2. BACKGROUND

Ad-Network with the use of a JavaScript code which is embedded within the Publisher’s source code. When a user visits the platform of a Publisher, the embedded code forwards a request to the Ad-Network who selects an advert from one of the Advertisers and features it to the user within one of the Publisher’s Ad-Boxes.

The selection of the advert which is to be displayed by the Ad-network on each individual instance is performed through an operation called Advert Auctioning. To participate in an auction, Advertisers place bids which represent the price they are willing to pay to the Ad-Network for every time their adverts are viewed or clicked on by users. The bidding strategies which are applied by an Advertiser to estimate the value of a placed bid varies for each advert publication as it is typically determined based on contextual data of the particular Publisher and user. For example, an Advertiser who promotes holiday deals may be willing to offer a higher bid when their advert is displayed on the platform of a Publisher who offers tourist information and is being displayed to a user who is near an airport. Once the bids from all participating Advertisers have been placed, the Ad-Network ranks the candidate adverts based on their potential profitability. To calculate the rankings, the Ad-Network applies a formula which combines the bid value of each candidate advert with a quality score that expresses the likelihood of set advert being relevant to the user’s interests. The quality score of each distinct user is determined by the Ad-Network based on contextual attributes (e.g., age, gender, location and search history) which are obtained by tracking the user’s activities and is determined based on statistical data. Upon completing the auction, the Ad-Network declares the highest ranking candidate advert as the winner and features it to the user within the Publisher’s Ad-Box.

Regarding the rewards which are issued to the participating stakeholders for their services, the OBA system supports three pricing models. The first and most simple model is referred as Pay-Per-Mille or PPM and is founded on the principle that an Advertiser awards the Ad-Network for every one thousand times their advert is viewed by a user (hence the term 'mille' which is Latin for 'thousand'). The second model is called Pay-Per-Click or PPC and is used to award Ad-Networks every time one of the adverts they illustrate is clicked by a user. The third and final model is known as Pay-Per-Action or PPA and awards Ad-Networks when the user who clicked on an advert also performed a specific action. Most typically this action is the completion of a purchase or the creation of an account [54]. Based on the pricing model which is enforced, the Ad-Network can claim their reward by filing an Ad-Report to the corresponding Advertiser and afterwards awards
a commission to the Publisher for providing the Ad-Box.

The interaction between Publishers and Advertisers takes place in real time and is facilitated by the infrastructure of the Ad-Network. The main component of the Ad-Network is the Real Time Biding (RTB) platform which supervises the Advert Auctioning and establishes a communication link between Publishers and Advertisers. The RTB platform may also incorporate additional modules for assigning, matching and storing the contextual attributes of the Publishers and the users. Publishers connect to the RTB platform through an Ad-Network provided panel (e.g. Google AdSense) which manages the inventory of available Ad-Boxes, sets the parameters of each entry (media format, price floor, pricing model, etc.) and provides sale feedback. In a similar fashion, the Advertisers access the RTB platform via their own Ad-Network provided panel (e.g. Google AdWorks) which is used to store adverts, run bidding strategies and manage advertising campaigns. Alternatively, the Publishers and Advertisers can connect to the Ad-Network via third-party platforms which are respectively known as the Supply Side Platform (SSP) and the Demand Side Platform (DSP). The main advantage of the use of SSP and DSP is that they allow the Publishers and Advertisers to simultaneously connect to multiple Ad-Networks, thus establishing and extended advertising market which is known as the Ad Exchange.

2.2 Consumer Tracking

In order to determine the advertising needs of consumers, OBA relies on tracking technologies which collect and analyze vast amounts of data across multiple platforms. The most prominent types of information which is typically collected by Ad-Networks and the ways it can be exploited for advert targeting are analysed in the following paragraphs.

Demographics: Demographic data is defined as a set of factors that express the socioeconomic characteristics of an individual [39]. This may include traits such as age, gender, ethnicity, spoken language, religion, marital status, occupation and income. A user’s demographics can be inferred by their social media profiles, service registration forms and other online activities. Demographic data is typically used to segment users into consumer groups which are associated to specific interests. For example, female consumers of higher income may be associated to adverts for designer clothes.
**Browsing and Search History:** A user’s browsing and search history can be tracked primarily through the use of technologies such as web cookies, browser fingerprinting and third-party domain requests. Cookies are files that are downloaded from browsed websites and remain locally stored on the user’s machine until they can be sent to the Ad-Network via another domain that is visited by the user [89]. Browser fingerprinting is a technique for identifying users based on the unique configurations of their browsers (e.g., languages, fonts, extensions, etc.) [18, 42]. Third-party domain requests are sent by a browser in order to obtain elements which are embedded in the code of a visited website but are stored in a different domain (e.g., banners, media files, social network share buttons, etc.) [103, 118].
search history logs are particularly useful to marketers as they can reveal a user’s consumer interests. For example, a user who performs a search for holiday destinations can be assumed to be interested in traveling and targeted with adverts of airline companies. Browsing and search logs can also be used for even more advanced targeting strategies such as re-targeting. Re-targeting is a fine-grain targeting method which is used to present users with adverts of specific products that they have previously browsed. As an example of re-targeting consider a scenario where a user visits the website of a retail store, navigates through the catalog of female clothing and selects a jacket. When the user navigates to another website, she will not simply be presented with an advert which is relevant to female clothing but rather an advert that depicts the exact same jacket that she previously selected for viewing.

**Purchase History:** A user’s purchase history can be used for the promotion of products which are relevant or supplementary to the user’s previous purchases. For example, a user who bought a pair of sports shoes may be targeted with adverts for other sporting equipment and a user who purchased a car may be targeted with adverts by insurance companies. Purchase history information is mostly used by online retailers such as Amazon and eBay but it can also be exploited by other marketers through the use of credit services such as Apple Pay, Android Pay and PayPal.

**Social Media Publications:** The information that is shared on social media is often exploited for advert targeting as it offers a very detailed insight of a user’s consumer preferences. Other than providing demographic data (age, gender, occupation, etc.), social media publications can also reveal personal affiliations, attended events, visited locations and interests for specific activities or brands. For example, a user who has attended events at Stamford Bridge stadium and is a member of a Facebook group that is dedicated to football may be targeted with adverts for Chelsea FC paraphernalia while a user who follows Apple on Twitter can be targeted with adverts for the latest iPhone.

**Location Data:** Location Based Advertising or LBA is a marketing method for promoting location-specific products and services [87]. Location data can be obtained primarily via GPS but can also be revealed by the IP addresses of WiFi hot-spots that the user connects from, social media check-ins or even Bluetooth Low Energy (BLE) beacons. As an example, consider a sce-
nario where a user with a Bluetooth enabled device appears within range of a BLE beacon during lunch time. The signal which is transmitted by the beacon is picked up by the device which is directed to a cloud server in order to download adverts for restaurants within the user’s vicinity.

**Hardware and Software Data:** Hardware and software data may reveal the make and model of a user’s device, the operating system that is run, installed apps or even connected peripheral devices. The simplest way to exploit hardware and software information is by identifying the electronic devices that the user is already employing and target him/her with adverts for relevant products. A simple example of the aforementioned targeting method would be to show adverts for a new Apple product to a user who is accessing the web via iOS. Additionally, marketers may also infer other information about a user based on their hardware and software. For example, owning an expensive premium device can reveal high financial status or using a VR headset and a game-pad controller may indicate an interest for mobile gaming.

**Linguistic Data:** Linguistic data refers to the written information that users share in mediums such as emails, instant messages, calendar entries and social media posts. Ad-Networks and market researchers have been known to exploit linguistic data with a technique called Opinion Mining. Opinion Mining is a method for uncovering opinion-oriented information from written text [115]. Opinion Mining is made possible through the application of Natural Language Processing (NLP) which is the research area that explores how computers can be used to understand and manipulate natural language for the purpose of performing useful tasks [29]. An example of the use of Opinion Mining in marketing would involve the targeting of a user with hotel adverts after they have sent an email where they express their intention of going on a holiday.

**Sensor Data:** Sensor data is any information which can be gathered by smart devices such as health monitors, voice controllers, network cameras, programmable dash buttons or other gear which is integrated to the IoT (Internet of Things). The Internet of Things is a relatively new technology but despite this, IoT devices are already incorporated into targeted advertising with the most notable example being the use of Amazon’s virtual assistant for the gathering of consumer information [65, 136].
2.3 Privacy Concerns and Countermeasures

OBA can be stated to be mutually beneficial for all participating parties, including the users who not only get informed about products that are relevant to their needs but also enjoy free Publisher services. Nevertheless, the current implementation of the OBA model is far from ideal as it also presents serious privacy concerns. The private data that is collected by Ad-Networks may reveal sensitive user information such as income, health status, sexual orientation, religious beliefs, political and social opinions, ideological inclinations, location patterns, shopping and browsing habits and lifestyle preferences.

The intrusive information gathering which is conducted for the purpose of advert targeting has provoked many users to adopt countermeasures such as ad-blockers, cookie managers and obfuscation software. Ad-blockers such as AdBlock [46], Ghostery [6], uBlock [9] and Disconnect [2] have seen a notable rise in popularity over the past years [113]. Ad-blockers are security software tools for preventing third-party domain requests [152]. The use of third-party domain requests is one of the primary methods to obtain a user’s browsing history, as outlined in Section 2.2. Cookie managers are security tools which are used to remove web cookies. Cookie managers are typically implemented as browser add-ons or can even be integrated in security-oriented browsers such as Brave [7], Firefox Focus [5], Epic [4] and Yandex [3]. Obfuscation software applications are privacy tools which obscure the user’s activity by generating fake information with the intent to produce noise. Some prominent examples of obfuscation tools include the following: TrackMeNot [8] is a browser add-on which performs random searches in order to confuse search engines about the user’s real search queries. AdNauseam [1] follows a similar approach to TrackMeNot [8] but focuses on misleading Ad-Networks by automatically clicking on web adverts. NOYB [68] fills out registration forms with fake data which is indistinguishable from real entries. The real data can still be obtained by authorized users through a cryptographic process. ProfileGuard [140] is a mobile obfuscation tool for preventing Ad-Networks from profiling users based on the apps they download on their mobile devices. Lastly, MockDroid [17] is a modified version of the Android operating system which allows users to input fake system information (e.g., device ID, GPS location, internet access, etc.) on the installed mobile apps.
CHAPTER 2. BACKGROUND

2.4 Advertising Fraud

Advertising fraud is committed by malicious Publishers who are able to claim service commissions for artificially created views and clicks on their platforms. Some of the most prominent methods of advertising fraud are considered in the following paragraphs.

Keyword Spamdexing: Keyword spamdexing, also known as keyword stuffing, is the practice of repeating certain keywords within the text of a website for the purpose of misleading web crawlers [148]. Web crawlers, also known as spiders or spiderbots, are automated programs that are tasked with classifying the content of websites for the purpose of search engine indexing. By engaging in keyword spamdexing, a fraudulent Publisher can deceive Ad-Networks into over-evaluating her platform during an Advert Auctioning [132]. Consider a simple example where an Advertiser that promotes plane tickets participates in an Advert Auction. Based on her bidding strategy, the Advertiser typically places a bid of $1 for an available Ad-Box but is willing to raise the bid to $1.5 when the Ad-Box appears within a website that relates to traveling. To trick the Advertiser into placing a higher bid, the malicious Publisher can misrepresent the content of her website by stuffing it full of keywords which relate to travel (e.g., holiday, vacation, trip, hotel, etc.). To avoid raising the suspicion of visitors, the Publisher can place the keywords within HTML tags or make them illegible by using text that is too small to read or text that is the same color as the website’s background [132].

Advert Placement Fraud: Advert placement fraud, also known as advert farming, is carried out by a rogue Publisher who loads a score of advert banners which are not visible to the user [95]. To implement fraud of this type, Publishers can stack multiple adverts on top of each other, perpetually reload advert banners, open hidden pop-under windows or change the size of advert banners which are sometimes made as small as a single pixel. An indicative example of advert farming is disclosed in [43] where the authors proclaim the discovery of a fraudulent Publisher called Hula. Similar findings are presented by Liu et al. [96] who were able to uncover more than a thousand mobile apps that were committing fraud by misplacing or modifying advert banners.

Coercion: Coercion is performed by Publishers who dupe users into clicking on adverts for reasons other than their content [132]. To perform this
type of fraud, Publishers can disguise adverts as links, make them invisible or embed them within other elements of the website such as flash games [34].

**Forced Browser Clicks:** Forced clicks can be artificially performed by a browser without the user’s knowledge with the use of a client-side script [132]. The script is inlaid within the Publisher’s source code and perceived as a legitimate part of the visited website. Gandhi et al. [63] demonstrate such an attack where the functionality of the JavaScript which is used by the Ad-Network to load adverts is altered by a second malicious JavaScript. By carrying out the aforementioned attack, the authors were able to simulate clicks even on adverts that were placed on a different domain than the one viewed by the victimized user.

**Manual Clicking:** Performing manual clicks is a simple and low-tech method for artificially generating advert revenue. For maximized effectiveness, Publishers can employ the services of click-farms. Click-farms are facilities, typically in developing countries, that accommodate groups of low-paid workers who click on digital adverts. Depending on the scale and the rate at which the fraud takes place, forged clicks may potentially be detected by Ad-Networks through policy-based filtering. As a means of avoiding detection, fraudsters can use proxies to hide their source IP addresses and also obfuscate their activity by mixing it with the traffic of other users.

**Auto-clickers:** Auto-clickers are simple programs which can be used to automate the clicking process on the system they are installed. Auto-clickers are readily available and easy to operate but suffer from the same shortcomings as manual clicking. Namely, auto-clickers produce a large volume of traffic which can easily be detected by Ad-Networks. Some advanced auto-clickers, which have been specifically designed for click fraud, may also incorporate proxy connections as a means of covering their tracks.

**Clickbots:** Clickbots are malicious software programs which perform automated clicks on adverts. Unlike auto-clickers which are intentionally installed by the operator of a system, clickbots run on infected machines while the victimized users remain unaware. Multiple infected machines can interconnect to each other to form a botnet which can be coordinated to launch large scale fraud. Clickbots are notoriously difficult to detect as they often employ very sophisticated methods in order to mimic real user activity. A notable example of how elaborate clickbot activity can be is Methbot.
CHAPTER 2. BACKGROUND

Methbot was discovered by White Ops in 2016 and is thought to be one of the biggest and most profitable click-fraud operations to date with an estimated loot of 3 to 5 million US dollars per day. To avoid detection, Methbot performed mouse movements and clicks, faked social media log-ins, manipulated geolocation information and employed specialized countermeasures against the code from over a dozen ad-tech companies [150].

2.5 Advertising Fraud Detection

To detect instances of advertising fraud, Ad-Networks rely on filtering systems which analyze the traffic patterns of Publisher platforms for suspicious activity. What constitutes as suspicious activity is any abnormality on the platform’s qualitative and quantitative metrics. Some prominent examples of traffic abnormalities which may reveal fraudulent activity are examined in the following paragraphs.

Repetitive Source IP Addresses: Source IP addresses which appear repeatedly can be attributed to clicks from the same network. When the clicks take place between regular or unrealistically short intervals of time, it can indicate the operation of an auto-clicker or a simplistic clickbot. Activity of this type is relatively easy to detect and block. On the other hand, clicks that take place between longer and more irregular intervals of time may originate from a manual clicker or a sophisticated clickbot that runs on an infected machine. Identifying fraudulent activity of this type is much harder and resource consuming as it requires the examination of large segments of data streams (often called ‘windows’ in data mining).

Non-residential IP Addresses: Legitimate users are typically expected to be accessing the web from private connections which are allocated residential IP addresses. In that regard, activity from commercial IP addresses can signify the operation of a data-center proxy which is used to hide the real source of the traffic. Identifying source IP addresses which are registered as commercial is straightforward but it may not always be enough as commercial IPs are also utilized by users who employ the services of a VPN provider.

Obscure Publisher listings: According to marketers, the majority of advertising fraud originates from obscure websites while real consumers mostly view adverts on well-known Publishers [101]. By auditing the listings of
reports, Ad-Networks may potentially be able to recognize suspicious Publishers and blacklist them or submit them to further investigation.

Traffic Spikes: A sudden increase in views or clicks may be triggered by the operation of an auto-clicker or unsophisticated clickbot. Rapid traffic changes are easy to recognize, even by the analytics software of the Advertisers [100].

Inflated or Similar CRTs CTR (Click-Through Rare) is a metric that expresses the ratio of clicks on a specific advert to the number of views. Unusually high CTRs can be a tell-tale sign of fraud while CTRs that share similar values can be the result of an auto-clicker or clickbot.

Poor or Repetitive Clickstreams: A clickstream is a virtual record of a user’s internet activity. Clickstrams include information such as the websites the user has visited, how much time was spent on each site, any data that was downloaded or uploaded (media files, messages, log-ins, etc.) and even the emails that the user has exchanged with other users. Clickstreams that show minimal user activity beyond the clicking on adverts are likely to originate from auto-clickers or click-farms. Similarly, clickstreams that exhibit repetitive and formulaic activity may be the result of elaborate clickbots which have been designed to mimic human behaviour based on scripted instructions.

2.6 OBA Issues and Solution Approaches

The operational practices of the currently implemented OBA system greatly threaten the interests of both users and Advertisers. For users, the intrusive collection of personal data by Ad-Networks creates a serious privacy threat which has provoked the adoption of tools that endangers the financial viability of the online advertising market. The risk to user privacy is further increased by the proliferation of cloud services which promotes the concentration of data within the care of a limited number of cloud providers thus creating a single point of failure. A preferable alternative which can potentially reduce the privacy threat of consumer data collection is the adoption of client-side processing which enables end users to locally process their own data. In the context of advertising, client-side processing can be exploited to track a user’s activity and determine their advertising interests without any data ever leaving the user’s smart phone. Client-side process is one of
the prime technologies that forms the basis of privacy-preserving advertising systems as further discussed in Section 3.1.

The ever-increasing expansion and the sophistication of advertising fraud call for the deployment of more effective countermeasures. The data filtering methods which are presently being deployed by Ad-Networks for the detection of invalid traffic have been proven to be incapable to rise up to the challenge [88], thusly spawning a dire need for the development of alternative designs. To address the new challenges of advertising fraud, researchers have shifted their focus to new and innovative technologies which are further detailed in Section 3.2
Chapter 3

Related Work

In this chapter we provide a detailed consideration of the related work in the research area of advertising. The chapter is separated into three distinct sections. In Section 3.1 we examine privacy-preserving advertising systems and establish the four main approaches which are typically used to ensure privacy. In Subsection 3.1.1 we scrutinize the most prominent designs and in Subsection 3.1.2 we proceed to identify their limitations. In Section 3.2 we focus on the field of advertising fraud. In Subsection 3.2.1 we analyze the available fraud prevention mechanisms and in Subsection 3.2.2 we point out their respective limitations. Finally, in Section 3.3 we perform a critical summary of our literature analysis and specify how our design expands upon the existing work.

3.1 Advertising Privacy

Providing privacy for targeted advert delivery models is a research problem that several academics have attempted to resolve. Most researchers agree that sensitive user data should be kept outside the reach of the Ad-Network, Advertisers, Publishers, and any other party that is not considered trusted. To address this problem, previously proposed models have incorporated various combinations of architectures and privacy mechanisms which are briefly summarized in the following paragraphs.

**Trusted Proxy** The simplest method of achieving anonymity is by introducing some form of trusted third party that acts as a proxy between users and the Ad-Network. The role of the proxy is to mask the identity of the user by forwarding requests after replacing any identifying information with
a temporary identifier. The *Ad-Network* uses this temporary identifier to reply to the proxy with relevant adverts which are then conveyed back to the user. In order to further increase privacy, public key encryption can be used to encrypt requests and adverts. When paired with cryptography, the proxy is aware of the identity of the user that is sending a request but cannot see the content of the requests or the corresponding adverts. In turn, the *Ad-Network* can read the encrypted requests but is not aware of the user’s true identity as it is masked by the proxy.

**Pool-of-Ads** Schemes that are based on the pool-of-ads approach make use of client-side processing to allow users to select adverts that best satisfy their needs out of a wider collection (pool) of available content. The pool can be populated by various methods with the simplest one being by making a generic request. When following this approach, a user issues a request for adverts that fall under a very broad category which includes their specific interest. For example, if a user is interested in running footwear they may make a generic request for sporting equipment. The *Ad-Network* responds with multiple adverts that satisfy the request and it is up to the user to keep the ones that best match their particular interest and discard the rest.

**Anonymous-Download** Schemes which use this approach enable users to directly download broadcast adverts through the use of specialized hardware and software. Advertisers store their adverts at broadcasting stations that operate in publicly accessible locations. As a user comes into proximity of these stations, their device downloads the available adverts. The user’s device is then responsible for sorting through the collected adverts and selecting the most relevant while the rest are discarded. Anonymity is achieved through the use of protocols that enable mobile devices to connect to the broadcasting stations without disclosing any information that exposes the user’s identity such as username or network address. Some of these systems also enable users to connect and exchange adverts with each other. In these systems, a user downloads adverts from a broadcasting station and then propagates them to other users that they later come into proximity with. This extends the reach of the broadcasting stations but it also requires a certain level of trust among users.

**Aggregation** Systems that incorporate this approach accumulate the requests of multiple users with the intent to obfuscate the interests of each participant within the aggregate. The requests are then placed either by a
single user or a trusted proxy to the Ad-Network. The Ad-Network responds with the appropriate adverts which are distributed to the requesting users. The Ad-Network is able to learn the interests of the entire group of participating users but cannot distinguish the preferences of any specific user within the group. Depending on the enforced level of security, such systems may also incorporate cryptography or mix networks in order to offer additional privacy between users.

3.1.1 Literature Review

MyPULSE [38] is a client-server mobile application that delivers specialized advertisements by utilizing contextual information. The GPS coordinates of a client are associated with a local ZIP code which is sent to a server who responds with location-relevant adverts. The server does not obtain the exact position of the user but is still aware of the general geographical area as well as other consumer interest information. This architecture is easy to implement and adopt but offers limited effectiveness as it primarily focuses on location based adverts.

P2PMM [134] relies on a trusted proxy that is referred to as the Intermediary Services Provider (ISP). The ISP is entrusted with storing the user’s sensitive information and directly answering requests with adverts that are provided directly from merchants. P2PMM offers adequate targeting effectiveness but heavily relies on the integrity of the ISP in order to provide privacy. Although the ISP has no immediate interest to expose any information to the merchants, the existence of a single point of failure presents a notable risk.

Juels [80] also uses a proxy, that is called the Negotiant and is responsible for matching user profiles to a specific set of adverts. The adverts are then aggregated and posted on a bulletin board where they can be answered by advertisers. The work presents a variety of different schemes which account for different levels of security by utilizing methods such as public key cryptography and mix networks\(^1\). The system offers substantial privacy against advertisers but assumes the integrity of the Negotiant which presents a potential threat.

Tran et al. [138] make use of a hybrid approach which combines client-side processing and a trusted proxy for re-targeting. Re-targeting is a marketing strategy which is used to present consumers with specific products that they have viewed in the past. The system makes use of a decentralized

\(^1\)Mix networks are protocols which obfuscate the origin of a message by forwarding it through a chains of proxies.
architecture where the Ad-Network is broken down into an Ad Exchange (ADX) service and a set of Retargeters. Users are responsible for determining their own re-targeting preferences which are forwarded to Retargeters via the ADX in encrypted format. The Retargeters then place a bid to the ADX for an available Ad-Box on a Publisher’s platform and the Retargeter that wins the auction publishers their advert via a trusted proxy. Provided that the proxy is not compromised, the ADX only sees the placed bids and not the user’s re-targeting preferences or the published advert.

MoMa [20, 19] is a proxy based advertising system that also makes use of a commonly accessible bulletin board. Users create a series of orders based on a hierarchically organized catalogue of products and services. A trusted party is then used to post the user’s orders on MoMa. At the same time, advertisers also publish offers of their products based on the same catalogue. The system detects matches between orders and offers and contacts the trusted party which notifies the appropriate user. MoMa presents a decentralized architecture which is appealing from a practical point of view but relies heavily on a trusted proxy in order to ensure user privacy.

Privad [66, 67] performs a selection from a pool-of-ads but also incorporates a trusted third party (the Dealer) to operate between the users and the Ad-Network. Unlike MoMa [20, 19] that operates with orders from a specific catalogue, Privad allows the user to select a general interest category and send it to a Broker (system equivalent of the Ad-Network) through the Dealer. Upon receiving the message, the Broker uses the same path to respond with a pool-of-ads which are relevant to the user’s interest. The user then sorts through the delivered pool-of-ads and selects the most prominent to be displayed while discarding the rest. Although this method is simple to incorporate into the existing model, it assumes the existence of a fully trusted third party to act as the Dealer. The Dealer is also a single point of failure and, if compromised, the security of the entire system can be bypassed. A fully functional prototype of Privad was constructed and tested in [121]. The results identify some limitation of Privad in terms of profiling and anonymity when dealing with a small number of users. Regardless of this, the work determines that a proxy-based solution is effective and practically feasible. The research outlined in [122] expends the capability of Privad by proposing an auctioning mechanism for privately ranking and calculating the cost of adverts.

ObliviAd [15] follows a similar architecture as other proxy-based systems but with the use of a secure hardware device that is placed on the Ad-Network side. The device receives requests from clients and responds with
a number of matching adverts that are obtained from the Ad-Network’s database. The system accounts for click reports and maintains privacy by deploying a Private Information Retrieval (PIR) mechanism which allows the client to access the Ad-Network’s database, while preventing the Ad-Network from learning about the query and the resulting answer [28]. Aside from requiring additional computational power, this architecture does not guarantee that the operator of the Ad-Network will not bypass security by physically tampering with the device.

RePriv [60] is a browser extension that offers effective advert delivery but also addresses the issue of privacy. The application tracks the user’s activity and builds a dynamic profile of consumer interests which are shared with advert providers. Privacy is enforced by allowing the user to explicitly give consent for what kind of information is disclosed to each party. This direct security approach is straightforward and easy to implement but is also very impractical for users. This may potentially make RePriv ineffective as frustrated users are likely to simply decline the sharing of all data.

MoRePriv [36] is an evolution of RePriv [60] with a special focus in mobile advertising. MoRePriv is implemented as part of the mobile Operating System (OS) and allows users to manage the sharing of private information in a similar manner as location data is managed by current systems. Additionally, MoRePriv offers support for private targeting by associating the advertising habits of users to a predetermined set of typical consumer profiles which are referenced as Personae. The system supports a total of eight Personae which can be associated to specific advertising interest in a similar fashion as demographic data is being utilized in traditional targeting. Although the system offers some flexibility by assigning a weight to each Personae in order to signify the extend in which it relates to a user, the limited number of Personae result in fewer options for targeted advertising.

Leontiadis et al. [90] propose a framework to control the stream of private information which is disclosed to the Ad-Network. Contrast to RePriv [60] which requires users to determine what kind of information is shared, Leontiadis argues that the flow of information should not be left in the hands of the user. Instead, the model takes a market-oriented approach in order to balance the flow of private information in accordance to the generated effectiveness. Consequently, access to private data is allowed to Publishers (mobile apps) which generate a low CTR (Click Through Rate refers to the ratio clicks per advert impression) while Publishers with high CTR are restricted in the amount of data they have access to.

Hardt et al. [72] argue that the optimal trade-off between targeting effectiveness and privacy can only be achieved through a hybrid system which
combines client-side profiling with a trusted proxy. The authors propose a framework which allows users to control the flow of private information which is shared with a partially trusted server. Based on the provided data, the server is able to respond to general interest requests with a pool-of-ads which satisfy the user’s request. The user is then tasked with selecting the most prominent advert and discard the rests. The multilayered architecture of the system does increase privacy but the use of a pool-of-ads creates unnecessary overhead and may also not be entirely effective when the user’s request is too generic.

Adnostic \cite{137} composes a local profile which is not disclosed to any outside parties but is used locally for the purpose of making a selection out of a pool-of-ads. When a user visits a website, a number of adverts that are relevant to the contextual theme of that particular website are sent by the Ad-Network. The advert which is the most relevant to the user’s interest profile is then selected and displayed while the rest are discarded. This approach is wasteful on resources and also offers limited targeting effectiveness as the delivered adverts are based on a very general assumption of the user’s interests. The system also allows the reporting of viewed adverts with the application of homomorphic encryption which is implemented with the assistance of a trusted party. The level of privacy therefore hinges on the integrity of the trusted party that manages the cryptographic keys. Should the Ad-Network gain access to advert reports, then user privacy is compromised despite the method which is used for advert delivery.

Kodialam et al. \cite{86} also follow a pool-of-ads approach which is similar to Adnostic \cite{137}. The authors propose a role reversal scheme where the Ad-Network sends to the user a series of interest profiles along with a set of matching adverts. The user stores the adverts that correspond to the profile that is the most similar to her own and the rest are discarded. This approach may potentially be more effective than Adnostic since users have a wider variety of adverts to choose from but the generated overhead is also increased to a negative effect.

BlueMall \cite{124} offers a rudimentary application of anonymous-download technologies. The model uses Bluetooth broadcasting in order to deliver adverts within a mall. Access points broadcast location aware adverts directly to user devices when they come into proximity. Although the downloading of adverts is private, the system also incorporates a central authority which keeps track of user location patterns and downloaded adverts. In terms of effectiveness, BlueMall is limited to local business and incapable of achieving fine-grain targeting.

PervAd \cite{23, 24} provides personalized adverts through broadcasting and
can also support fine-grained content. Users who maintain a local interest profile can collect relevant adverts as they move into proximity of customized WiFi access points. The system reduces overhead by first sharing some contextual information about the available adverts thus allowing users to selectively download only specific content. The interest profile is specified by the users themselves and the downloading process is performed anonymously. Even though this method achieves a substantial level of privacy, it is susceptible to malware as users have no means of verifying the integrity of downloaded content. Additionally, PervAd is also impractical as users need to manually compose their profiles and physically travel to specific locations in order to download adverts.

MobiAd [70, 71] is based on the principal idea of anonymous-download but also incorporates Delay Tolerant Networks (DTN). Users of the system maintain a local profile and are free to collect adverts via publicly accessible broadcasting stations such as Multimedia Broadcast and Multicast Services (MBMS) and WiFi hotspots. The system focuses on targeting effectiveness and therefore is limited to the collection of small volumes of fine-grain adverts. MobiAd also takes into account the issue of click report delivery by taking advantage of Delayed Tolerant Networks (DTNs) and public key cryptography.

Straub et al. [133] adopts a word-of-mouth approach in order to distribute adverts over a social network of peers. Users who appear within proximity of broadcasting stations are free to download adverts in accordance to a consumer profile which remains local. Additionally, users are able to also exchange adverts with each other by maintaining two lists which are referred to as 'iHave' and 'iWish'. The 'iHave' list stores the adverts which are held by a particular user while 'iWish' keeps adverts that the user wishes to obtain. To offer incentive for user participation, the system also includes a bonus point model which rewards users who propagate adverts. eNcentive [119] adopts a very similar approach to propagate adverts over an opportunistic network and also offers a rewarding method in order to provide user incentives. Both schemes provide privacy against Advertisers (local merchants) but for the most part ignore privacy concerns between the users. An additional factor that neither system does not account for is the presence of malicious users that may spread fake adverts or malware.

The Let’s Meet! [111] framework uses a client-server architecture which establishes a cooperation link between mobile users who share an interest for a particular offer but may be unrelated to each other. More specifically, Let’s Meet! enables consumers to take advantage of group offers by physically bringing them together in the location of a local vendor. The
authors emphasize privacy and security by incorporating mechanisms that prevent the disclosure of sensitive consumer information and defend against malicious users who may launch impersonation attacks or attempt to forge offer-coupons.

Artail et al. [14] and Fawaz et al. [58, 57] take advantage of user cooperation in order to achieve anonymity via aggregation. Nearby devices that wish to download adverts get connected and aggregate all of their interest profiles into a single device. The collective request is then forwarded to the Broker (system equivalent of the Ad-Network) who responds with the requested adverts. The received adverts are then delivered to the participants. The protocol achieves privacy against the Broker as well as other users by taking advantage of techniques such as mix networks and asymmetric encryption. This may put great strain to the hardware of the mobile device, especially if you take into consideration that the system operates in real time.

MASTAds [125] combines interest aggregation with the use of a partially trusted party that is called the AMS. The AMS tracks the contact patterns of users in order to divide them into communities. The interests of all members of a community are then aggregated and sent to the AMS which is responsible for fetching adverts from the Ad-Network. The relevant adverts are obtained by the AMS and are then propagated through the community via opportunistic networking. MASTAds reduces bandwidth and battery consumption and also has the prospects of achieving fine-grained targeting. The system also offers privacy among the users of a community and the Ad-Network but still relies heavily on the integrity of the AMS which has access to user meeting patterns.

Wang et al. [147] exploit the cooperation of users within a certain area as a means of delivering private location-based adverts. The model divides users into those who are sensitive about their location privacy (SUs) and those who are not (ISs). IUs are rewarded by assisting SUs within their vicinity to obtain location-relevant adverts. IUs download and broadcast a series of advert identifiers adIDs which are collected by nearby SUs. SUs select the adIDs which are relevant to their interests and forward them to a Publisher (content provider). The Publisher is then tasked to operate as a trusted proxy and present the requested adverts directly on his platform without exposing the users identity to the Ad-Network. This architecture offers a substantial level of privacy against the Ad-Network and other users but not against Publishers. Although this approach distributes trust to multiple Publishers rather than a single Ad-Network, the fact that Publishers have every incentive to expose user information to the Ad-Network presents...
a definite privacy risk.

The authors in [146] attempt to work around the problem of user privacy by approaching the advertising model from a game theory perspective. The proposed framework aims to motivate users to share their private data in exchange for monetary compensation from the Ad-Network. Their simulations demonstrate that when all parties are incentivized to actively engage in targeted advertising, the system eventually settles to an equilibrium which yields the optimal reward for the participants. The authors state that click fraud is assumed to be prevented by existing means. This may not be entirely accurate however since this design can serve as a catalyst for multiple malicious users to perform invalid clicks. Arguably, this will provoke a distributed form of fraudulent behaviour and make the problem of click fraud even more severe and harder to combat.

3.1.2 Limitations of Advertising Privacy Systems

Table 3.1 shows a detailed scrutiny of all of the examined privacy-preserving advertising systems. Our analysis focuses on three aspects, namely privacy, targeting effectiveness and practicality. Privacy indicates the system’s ability to protect the user’s private data from all other parties (including other users). Targeting effectiveness articulates the system’s ability to support fine-grained targeted adverts. Lastly, practicality expresses the level of practical difficulty which is required for the system to be implemented and maintained.

Systems which rely on anonymity proxies can be easily incorporated into the current architecture and achieve an adequate level of privacy without seriously reducing the targeting effectiveness and the efficiency of the system. Regardless, such systems assume the existence of a trusted or partially trusted third party that can act as the proxy. This assumption is not entirely realistic and also creates a single point of failure that threatens the integrity of the system if compromised. In essence, systems of this type do not really solve the problem but only transfer accountability to a different entity. The aforementioned limitations could potentially be resolved through the adoption of a proxy network such as TOR. By distributing trust across multiple proxies, the threat of a potential compromise is reduced but at the same time there is a significant increase in complexity and latency.

Performing a selection from a Pool-of-Ads maintains privacy by taking advantage of the computational capability of mobile devices. Systems which incorporate this design do not require any trust between the participants and, depending on the method that is used to populate the pool,
they can achieve a satisfactory level of privacy. Although these systems are not too difficult to introduce, they suffer greatly in terms of targeting effectiveness as the information that is shared with the Ad-Network may be too generic to effectively retrieve adverts that accurately correspond with the user’s interests. Additionally, a significant amount of unnecessary overhead is generated by these systems as multiple adverts need to be downloaded and stored only to be discarded afterwards.

Systems that incorporate Anonymous-Download technologies offer the highest level of privacy but may also be impractical due to the fact that users need to physically travel to designated locations where broadcasting station are accessible. When enhanced with opportunistic networking, the practicality of Anonymous-Download systems may increase but this will also raise trust issues among the nodes of the network.

Systems which enforce the aggregation of adverts from multiple users offer variable levels of privacy depending on their design. Models which combine aggregation with a proxy are not entirely private as they assume the trust of a third party. On the other hand, models that only rely on user cooperation have the potential of offering adequate privacy against the Ad-Network but not against other users. Some designs attempt to mitigate the aforementioned shortcoming by incorporating cryptography and mix-networks but this has the negative side effect of also increasing complexity and overhead which is not ideal for real-time applications.

All of the systems that were examined, rely on user profiles that are stored either locally or on a trusted proxy. This design assumes that each user only connects through a single device and that multiple users do not have access to the same device. If a user were to connect through multiple devices, then each device would only have partial knowledge of the particular user’s advertising profile which could cause a significant reduction of targeting effectiveness. To overcome this limitation, a system would need to be able to track users across multiple devices which is not easily feasible without infringing user privacy. In a similar fashion, an advertising profile which is associated with a particular device would be skewed if the device were to be accessed by multiple users. This eventuality can however be avoided when the device is able to distinguish between different users and compose separate advertising profiles. Regarding the submission of Ad-Reports, none of the examined systems is able to offer an architecture which is resilient to fraud while maintaining user privacy and targeting effectiveness.


3.2 Fraud Prevention

A number of researchers have contributed different mechanisms with the aim to combat advert fraud. As there are many ways of approaching the issue, their work varies greatly. Some focus on preventing fraudulent reports by detecting and blocking them at their source while others attempt to filter out illegitimate reports by validating their quality after they have been submitted.
3.2.1 Literature Review

CCFDP [83, 145] offers real time click fraud protection capability through the fusion of data (evidence of suspicious behaviour) which is provided by multiple collaborating sources. Three modules are used to independently evaluate reported clicks from both the server and client side and individually return probabilistic estimates of a click’s legitimacy which are combined to produce an overall score. The results are shown to improve the quality assessment of incoming traffic by an average of 10% compared to what is separately achieved by the individual modules, thus allowing the system to identify sources of fraudulent clicks more accurately and successfully block them. CCFDP offers increased effectiveness in comparison to currently adopted systems but it also endangers user privacy as it requires the collection of data from both the server and the client.

Rather that filtering our fraudulent clicks, Juels et al. [81] promotes the use of premium clicks which represent reports from users whose legitimacy can be verified through the use of cryptographic credentials, simply known as Coupons. Designated websites, referred as Attestors, provide their visitors with coupons when they perform specific tasks which are indicative of real user behaviour (e.g., making an online purchase). The coupon can be then attached to future Ad-Reports and works as a form of proof that a particular click was performed by a verified user. The model is implemented in such a way that the users’ identity is substantially protected against a curious adversary and also offers protection against coupon-replay attacks. Although the system is highly effective, one potential limitation lies on the fact that the submission of numerous reports will require the same number of Coupons which may not always be easily available.

Haddadi et al. [69] argues that click fraud is progressively becoming harder to detect through traditional threshold techniques (identifying multiple reports from the same source) as botnet activity is becoming evermore sophisticated through the employment of such means as proxies and distributed attacks. To address the problem, the paper proposes the use of specialized adverts which are called Bluff-ads. Bluff-ads operate as a form of honeypot which allures automated clickers but repels real users. While most adverts are typically targeted at a specific user by being context-specific to a consumer’s profile, Bluff-ads are purposely designed to be entirely irrelevant to the user’s interests (e.g., an advert for female clothes that is shown to a male user). As Bluff-ads are of no real significance to the targeted user, when they are being clicked they serve as indicators of suspicious activity. Although this idea to be very creative, the fact needs to be stated that
Bluff-ads are unlike to be adopted as they take up valuable space which can be used for real (profitable) adverts.

FCFraud [77] runs locally on the devices of individual users as a means of preventing them from being part of a botnet. Botnets are groups of infected devices which are used to commit click-fraud by generating fake reports without the user’s knowledge. The model is incorporated into the operating system as an anti-malware software which monitors submitted click-reports to detect if they correspond to real activity (physical mouse clicks) or have been artificially created by a malicious software. FCFraud is shown to be highly effective at recognizing infected machines but is ineffective against actors who intentionally commit such as click-farms and auto-clickers.

Faou et al. [56] provide a detailed examination of a click-fraud malware called Boaxxe over a long period. The authors run Boaxxe in a controlled environment and managed to reconstruct a redirection chain which maps the path of different domains that the malware follows before been directed to the targeted Advertiser’s website. By representing this data in a graph, they were able to identify key actors who have a critical role in the scheme and target them more effectively with the intent of disrupting the malware’s operation.

Zhan et al. [157] offers a pair of algorithms, GBF and TBF, which can be used to detect duplicate clicks on data streams which make use of Decaying Windows. The Decaying Window approach is a data mining method which is based on the premise of separating a data stream into segments (windows) which are examined individually. The objective of their work is to optimize the identification of clicks which appear in multiple windows with the use of Bloom filters. Their designs are shown to significantly reduce memory consumption while achieving a low rate of false positives and zero rate of false negatives.

MAdFraud [32] is a tool for identifying apps that engage in fraudulent behaviour. The system adopts a sandbox approach to trigger fraudulent activity by emulating user behaviour. MAdFraud was able to identify multiple apps which conduct fraud either by submitting impressions while running in the background or by fabricating fake clicks. From their results, the authors infer that fraudulent apps exhibit sophisticated stealth mechanisms such as pacing the rate of reports or using different Ad-Networks. Such means of remaining stealthy allows apps to avoid detection from systems which rely on filtering analysis on the server side.

DECAF [96] is a software implementation which analyzes the structural layout of mobile apps in order to identify developer violations of the regulations which are promoted by the Ad-Network. Namely, apps which commit
fraud by mismanaging adverts. This includes practices such as altering banners, publishing multiple adverts within the same Ad-Box/Ad-Slot or triggering fake clicks by placing adverts under other visual element of the UI.

AdAttester [92] is a proposed advert report verification framework which is based on the use of a secure hardware extension named ARM TrustZone. The device is capable of monitoring a phone’s input and output by directly connecting to the touch sensor and display modules. This allows TrustZone to verify both impressions (advert views) and clicks by comparing the user’s touch input to the location of a displayed advert on the screen. AdAttester appears to be highly effective at detecting fraudulent behaviour but it also requires the use of custom hardware which is costly and not always available.

A similar approach to AdAttester [92] but without the use of specialized hardware is proposed by Cho et al. [27]. The authors perform an empirical study of click fraud by implementing their own malicious software called ClickDroid. As a countermeasure against automated clickers, they suggest tracking the user’s clicks from the touch sensor at the kernel level. This is achieved with the installation of a middle-ware which collects the sensor’s output and logs it into a separate file. The log can then be used to verify if a particular report corresponds to a physical click. Assuming that the middle-ware cannot be bypassed by sophisticated bots, the system is still be susceptible to click-farms.

Hua et al. [75] propose an alternative architecture with the existing stakeholders of the advertising ecosystem. The authors suggest that users play a more active role in the process by directly forwarding reports to the concerned Advertisers and Publishers. Both parties then anonymize and forward their data to the Ad-Network who is responsible for matching and awarding each report. Reports are encrypted in such a way that Advertisers have access to the clicked advert but not the identity of the Publisher who presented it. This enables Advertisers to directly check the legitimacy of each click but may also incentivize them to commit themselves fraud against Publishers by denying the validity of submitted reports.

### 3.2.2 Limitations of Fraud Prevention Systems

Fraud prevention systems can be loosely classified in two categories in terms of the approach which they follow. The first, and most prominent classification, focuses at detecting fraudulent activity by analyzing the traffic patters of user activity on the server side. The challenge which is faced by this approach is in regards to the bulk of available information which makes...
processing difficult and costly. Furthermore, systems that rely on filtering algorithms are not entirely effective against click-farms and sophisticated clickbots that mimic real user behaviour. Concerning privacy, set approach heavily relies on sensitive data and therefore constitutes as a threat for users. The second classification consists of mechanisms which are aimed at detecting malicious activity at its source. Such systems adopt a variety of means such as sandbox analyzers, honeypots, adware programs, secure hardware and digital certificates. Systems of the second classification may have an advantage over traffic filtering as they are cheaper to operate and generally require less or no private user information. Nevertheless, such systems have mostly been deployed in supplementary roles as they also exhibit serious drawbacks in terms of effectiveness and practicality. More specifically, sandbox analysis and honeypots are only effective at identifying malicious programs but have no means of preventing their use. Furthermore, certain sophisticated clickbots have been known to be able to detect the presence of sandbox analyzers and honeypots. Adware programs and secure hardware operate on the client side and are therefore only effective at preventing users from involuntarily installing malicious software. Against users who are not concerned with actively preventing clickbots and against operators who intentionally commit fraud, such apparatuses have no effect. Lastly, digital certificates offer promising potential at addressing advertising fraud. It needs to be mentioned however that, if not implemented correctly, digital certificates may be susceptible to fabrication and may also violate user privacy.

3.3 Related Work Summary

Our assessment of the related work, in both research areas of advertising privacy and fraud prevention, shows that previously proposed systems have notable shortcomings. Privacy-preserving advertising systems are typically based on a combination of four approaches: (1) Trusted Proxy, (2) Pool-of-Ads, (3) Anonymous-Download and (4) Aggregation. Regardless of the approach being used, all systems that were examined fail to achieve an acceptable balance between privacy, targeting effectiveness and practicality which renders them unsuitable. Fraud prevention systems adopt a wide range of approaches which may partially be effective against certain types of fraudsters but not against all. More importantly, none of the relevant research addressed how the adoption of fraud prevention systems will affect user privacy. Consequently, privacy-preserving systems rely heavily on the
collection of private data which renders them as a serious threat for users. To the best of our knowledge, none of the previously proposed designs offers a comprehensive solution that ensures effective advert targeting and protection against advertising fraud while still maintaining user privacy. ADS+R was designed to fulfill this role by offering both targeted advertising and fraud prevention without violating user privacy. ADS+R adopts elements from previous designs (e.g. client-side processing, anonymous broadcasting technologies) and expands on the available research by incorporating new features with the aim to offer a complete advertising solution which is mutually beneficial for all concerned parties.
Chapter 4

ADS: Advert Distribution System

The Advert Distribution System (ADS), which was published in [99], is a novel approach for distributing personalized adverts over a social network of mobile users. ADS takes advantage of anonymous-download technologies which enable mobile devices to download promotional materials via publicly accessible broadcasting stations. Previously proposed anonymous-download designs such as [124, 23, 24, 70, 71] and [133] have been shown to offer a substantial level of privacy against Ad-Networks but still suffer from serious limitations as previously discussed in Section 3.1.2.

ADS aims to overcome the limitations of previous designs by achieving a balance between privacy and practicality while maintaining fine-grained targeting capability. Our design specifically focuses on mobile advertising which is the most widely used form of advertising and also one of the most intrusive as it exploits sensitive information such as location patterns, app usage and smart-phone sensor data. We expand on previous work by fusing client-side processing and anonymous-download technologies with opportunistic networking and public-key encryption. In comparison to contemporary designs, the client-side processing capability of ADS allows for (1) fine-grained targeting which offers greater advertising effectiveness. The combination of anonymous-download technologies and opportunistic networking, which is also present in ADS, helps to achieve (2) greater user privacy against the Broker, Advertisers, Ad-Dealers and other users and at the same time (3) makes the system more resilient to fake advert injection and sabotage attacks. Lastly, the application of opportunistic networking (4) expands the reach of the system by allowing limited mobility users, who do not appear
within the vicinity of broadcasting stations, to still be able to transfer data via their neighboring nodes.

In the following sections of this chapter we offer a detailed presentation, analysis and evaluation of ADS. We define the system’s specifications in Section 4.1 and then offer a detailed overview of the system in Section 4.2. In Section 4.3 we summarize the protocol and lastly, in Section 4.4 we evaluate our design.

4.1 System Specifications

In the following sections we define the specifications of ADS. We begin by identifying the system’s stakeholders in Section 4.1.1 and determine the trust relations between them in 4.1.2. Lastly, we proceed to set the system’s functional requirements in Section 4.1.3.

4.1.1 Stakeholders

ADS consists of five stakeholders, namely users, Advertisers, Publishers, Ad-Dealers and lastly the Broker. The first three represent the same entities as those of the OBA system which was exhibited in Section 2.1. We restate that users represent consumers who view adverts on their mobile devices, Advertisers are promotional companies and Publishers are digital platforms which display adverts. The Broker is selected by the Advertisers as their trusted representative. As Advertisers are too numerous to operate independently while still remaining coordinated, they employ the services of the Broker whose job is to function as an administrative authority. Lastly, Ad-Dealers are local broadcasting stations who serve as communication gateways between users and Advertisers.

Users who appear within the proximity of an Ad-Dealer, send their requests which are forwarded to the Advertisers. The corresponding adverts are sent back from the Advertisers to the Ad-Dealer so they can be broadcast. The user who made the request is responsible for downloading the broadcast adverts while the remaining users within the area ignore the transmission. The role of Ad-Dealer may be cast to any regional entity with physical presence in publicly accessible areas. This may include shopping malls, WiFi hotspots and local businesses. Ad-Dealers are free to conduct their own business independently and do not need to coordinate with each other. The necessary hardware and software infrastructure is provided to Ad-Dealers by the Broker who serves as their administrative authority on behalf of the Advertisers. The users are the ones who select which adverts
they want to download and send their requests directly to the Advertisers through the Broker’s infrastructure. The Ad-Dealers are only responsible for hosting the Broker’s hardware and do not need to interact with the Advertisers. For users, the only precondition to participate in ADS is to own a smart-phone device which runs the required software while Ad-Dealers can be added or removed dynamically.

4.1.2 Trust Model

The Broker is employed by the Advertisers to operate as their representative in the system. Since the Broker does not receive a share of the advert revenue, she has no benefit from deceiving the Advertisers and can therefore be assumed as trusted. On behalf of the Broker, no trust is required towards Advertisers.

Ad-Dealers are supplied by the Broker with specialized networking equipment which is installed on site. Despite having no immediate benefit from undermining the system, Ad-Dealers have the potential of tampering with the Broker’s infrastructure. To ensure Ad-Dealer integrity, the Broker can enforce preventive measures similar to Point Of Sale (POS) system providers. Such precautions can include legal agreements and periodical hardware checks. Since Ad-Dealers are registered businesses, they can be assumed as unlike to engage in criminal activity that can easily be traced back to them. The Broker and Advertisers are therefore suspicious of Ad-Dealers but do not consider them malicious. On behalf of Ad-Dealers, no trust towards the Broker and Advertisers is required.

Users consider the Broker, Advertisers and Ad-Dealers as honest enough to provide them with legitimate adverts but also curious and very determined to obtain private user data. Users can therefore trust the provided material but are not willing to expose any information that can link to their true identity (name, address, banking details, etc.). Users are also very distrustful of each other despite being part of the same social group. Compromised users can potentially expose sensitive data of other users or propagate fake adverts. Additionally, compromised users can sabotage the entire system by attacking Ad-Dealers. Users are therefore considered as malicious by the Broker, Advertisers, Ad-Dealers as well as other users. Lastly, Publishers only associate with users and operate independently to the rest of the system. No trust is therefore required between Publishers and other stakeholders.

Table 4.1 shows the trust relations between the system’s stakeholders. The first column lists the system stakeholders and each line exhibits the
level of trust of the respective stakeholder towards the remaining entities. A 'Trusted' label indicates that a stakeholder can be trusted to not perform any action that undermines the system. This level of trust is only exhibited by the Advertisers towards the Broker. A 'Suspicious' label indicates that a stakeholder has no benefit from acting maliciously but she is still expected to provide proof of her integrity. This level of trust is exhibited by the Broker and the Advertisers towards the Ad-Dealers. A 'Curious' label indicates that a stakeholder is trusted to not cause harm (for e.g. spread malicious software) but cannot be trusted to handle private information. The 'Curious’ label is attributed to stakeholders that have no reason to undermine the functionality of the system as this would inevitably cause them direct financial or legal damage. However, the same stakeholders are still willing to mishandle private information and are also able to do so without exposing themselves. This level of trust is exhibited by the User towards the Advertisers, Ad-Dealers and the Broker. Lastly, a 'Malicious' label indicates that a stakeholder is expected to act with criminal intent. This level of trust is exhibited by all system stakeholders (including users) towards any compromised user who may attempt to steal data or sabotage the system.

<table>
<thead>
<tr>
<th></th>
<th>Advertisers</th>
<th>Broker</th>
<th>Ad-Dealers</th>
<th>Users</th>
<th>Publishers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advertisers</td>
<td>-</td>
<td>Trusted</td>
<td>Suspicious</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Broker</td>
<td>-</td>
<td>-</td>
<td>Suspicious</td>
<td>Malicious</td>
<td>-</td>
</tr>
<tr>
<td>Ad-Dealers</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Users</td>
<td>Curious</td>
<td>Curious</td>
<td>Curious</td>
<td>Malicious</td>
<td>-</td>
</tr>
<tr>
<td>Publishers</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 4.1: Table of trust relations between ADS stakeholders.

4.1.3 System Requirements

In this section we list the functional requirements of ADS in consonance with the trust model that we established in Section 4.1.2. The ensuing requirements will serve as the criteria upon with our system will be evaluated.

- **User anonymity against Ad-Dealers**: Users view Ad-Dealers (and by association the Broker and Advertisers) as 'Curious' of their private data. It should therefore not be a way for Ad-Dealers to obtain any information that links a user’s identity to their advertising interests.

- **User privacy against other users**: Users consider other users as
'Malicious'. The advertising interests of a particular user should therefore not be accessible to other users.

- **User security against malicious users**: Malicious users are not restricted to the collection of private data but may also attempt to actively harm legitimate users by injecting fake adverts into the system. *ADS* should therefore not allow malicious users to propagate any harmful content to their peers.

- **Robustness against sabotage attacks**: Users are viewed as 'Malicious' by the remaining stakeholders as a compromised user may attempt to sabotage the system. *ADS* should therefore be protected against any attacks which may be launched by malicious users.

### 4.2 System Overview

*ADS* establishes a communication link between mobile device users and *Advertisers* by combining anonymous-download technology and opportunistic networking as shown in Figure 4.1. The *Broker* initiates the operation of the system by collecting adverts from the *Advertisers* and supplying them to the *Ad-Dealers*. Users run specialized software which automatically determines their advertising needs and requests suitable adverts when they appear within proximity of *Ad-Dealers*. The adverts are stored locally in the users’ devices until they can be displayed by a *Publisher*.

Alternatively, users can connect to *Ad-Dealers* indirectly via *Agents* who are themselves mobile users within the same social network. *Agents* are highly mobile users who regularly appear within range of *Ad-Dealers* and can therefore contribute to their community by downloading adverts on behalf of other users. This architecture allows users, who do not enter the proximity of *Ad-Dealers* often enough, to obtain their adverts by exploiting the mobility of *Agents* within their social cycle. Additionally, the presence of the opportunistic connection boosts the system’s anonymity as *Agents* serve the role of a partially trusted proxy.

To establish an opportunistic connection with the *Ad-Dealers*, a mobile user, who will from now forth be referred to as *Requester*, sends a request message to the *Agent*. The *Agent* physically ferries the request message to an *Ad-Dealer* and collects the relevant adverts. The *Agent* can then forward the collected adverts back to the *Requester* the next time the two of them come within proximity. An *Agent* can serve multiple *Requesters* simultaneously. Furthermore, users can operate as both *Requester* and *Agent*
at the same time. This would involve a user Alice sending a request to Bob who then forwards it to a third user named Charlie. In this scenario, Bob is the Requester for Charlie but at the same time serves as the Agent for Alice. ADS can support interactions with multiple users but for the sake of simplicity we will just explore the most basic scenario where only two users (a Requester and an Agent) are involved.

Both requests and adverts are transmitted across ADS in encrypted format. The use of cryptography preserves Requester privacy and also prevents the injection of fake adverts by a malicious Agent. Furthermore, the system incorporates authentication protocols which prevent attackers from impersonating users or Ad-Dealers. The detailed technical operation of ADS can be broken down in four phases which are described in Sections 4.2.1 to 4.2.4.
4.2.1 Phase 1: Setup

The preparatory phase of the system involves the recruitment of Ad-Dealers and the installation of a software client on the user’s mobile devices. The client is responsible for determining the user’s advertising interests and also registering the user’s contacts with his/her peers. Users are identified by a unique User Identity Name $ID_u$ which is inputted on the client at the moment of installation. Additionally, users are required to select a secret Master Password $MPW_u$. The $ID_u$ and $MPW_u$ are used to identify and authenticate the users to each other and are kept secret from all other stakeholders. The technical details of the aforementioned operations are provided in the following paragraphs.

Ad-Dealer Recruitment: The Broker launches ADS by recruiting a network of Ad-Dealers who are identified by a unique reference number $Aid$ (Ad-Dealer Identity). The Ad-Dealers are then equipped with specialized networking hardware which can support anonymous-downloading. The Broker also supplies the Ad-Dealers with two cryptographic keys, $SysDK$ (System Decryption Key) and $ADK^{sig}$ (Ad-Dealer Signing Key). $SysDK$ is an asymmetric decryption key which remains private among the Ad-Dealers. The corresponding public Encryption Key $SysEK$ is pre-installed within a mobile client which is downloaded by the users. Users can encrypt their requests with $SysEK$ and privately send them to Ad-Dealers who are able to decrypt them with $SysDK$. The second encryption key $ADK^{sig}$ is meant to be used for authentication and must therefore remain private among Ad-Dealers. The corresponding public key $ADK^{ver}$ is known as the Ad-Dealer Verification Key and is also pre-installed in the user’s mobile client.

Advertising Interest Profile (AIP) Composition: Mobile devices have access to a multitude of user information such as browsing logs, search history, emails, text messages, mobile app data, GPS and WiFi access-point locations, purchase logs (via services like Apple Pay or PayPal) and also data from on-board sensors such as accelerometers, pedometers and activity monitors. ADS taps into a smartphone’s resources and utilizes client-side processing in order to determine a user’s advertising needs with much higher effectiveness than a remote observer. The AIP is updated automatically based on the changes of the user’s activity. Similar approaches have been applied in the past by various systems such as [66, 67, 36, 72, 137, 70, 71] and [133].

Users join ADS by downloading and installing a mobile software client
which is tasked with constructing the user’s $AIP_u$ (Advertising Interest Profile). The $AIP_u$ is standard throughout the entirety of the system and represents a list of common consumer interests (automotive, technology, food and drink, etc.) which can be marked in binary format as 'TRUE' or 'FALSE'. To refer to each contained interest, $ADS$ uses a unique identifier $I_{id}$ (for e.g., $I_1, I_2, ..., I_n$). When given the right permissions, the mobile client can determine the user’s advertising interests and mark them as 'TRUE' on the $AIP_u$.

The $AIP_u$ is not shared with any other parties but remains local where it can be dynamically maintained based on changes in user behavior. The process which is used to deduce the user’s interests is independent to the rest of the system, thus offering a great deal of versatility. Individual $Advertisers$ would be able to fine-tune $ADS$ by implementing their own proprietary tracking algorithms which will be fully compatible with the rest of the system for as long as they produce an output that follows the standard $AIP_u$ format.

**Contact Registration:** In addition to composing the $AIP_u$, mobile clients also perform periodic scans (e.g., via Bluetooth or WiFi) in order to record the user’s encounters with his/her peers as well as with $Ad-Dealers$. When two mobile users appear within range for the first time, both clients request a manual confirmation to exchange $ID_u$s. Once confirmation has been achieved, the users generate a pair of $Contact Authentication Passwords$ $ConPW_{u1}^{u2}$ and $ConPW_{u2}^{u1}$. To generate $ConPW_{u1}^{u2}$ and $ConPW_{u2}^{u1}$ each user combines their own $Master Password$ $MPW_{u1}$ or $MPW_{u2}$ with the other user’s $User Identity Name$ $ID_{u1}$ or $ID_{u2}$. The results are then individually hushed to produce $ConPW_{u1}^{u2}$ and $ConPW_{u2}^{u1}$ as shown in Equations 4.1 and 4.2. Once successfully generated, $ConPW_{u1}^{u2}$ and $ConPW_{u2}^{u1}$ are exchanged and the two users are registered as $Contacts$. Ideally, an out-of-band channel (SMS, QR code, email or keyboard input) should be used for the exchange but alternative channels such as Bluetooth or WiFi can also be used. Although the use of an insecure channel simplifies the exchange, it also creates the risk of passwords being sniffed by nearby eavesdroppers. The compromise between security and utility can be left to the discretion of the user.

$$ConPW_{u1}^{u2} = h(ID_{u2}, MPW_{u1})$$  \hspace{1cm} (4.1)

$$ConPW_{u2}^{u1} = h(ID_{u1}, MPW_{u2})$$  \hspace{1cm} (4.2)

When the two registered $Contacts$ meet again in the future, they can log
their encounter \(^1\) after they have both verified each other with a challenge-response password handshake as depicted in Figure 4.2. The handshake begins with the two users \(u_1\) and \(u_2\) (1) exchanging their User Identity Names \(ID_{u_1}\) and \(ID_{u_2}\) as well as two random nonce challenges \(R_1\) and \(R_2\). (2) The users then calculate \(ConPW_{u_1}^{a_2}\) and \(ConPW_{u_2}^{a_1}\) in the same way as when they first registered as contacts. Alternatively, the users can keep Contact Passwords stored. This will reserve processing power at the expense of memory. (3) Users then calculate two Temporary Passwords \(P_1 = h(ConPW_{u_1}^{a_2}, R_2)\) and \(P_2 = h(ConPW_{u_2}^{a_1}, R_1)\) and (4) exchange them. Once \(P_1\) and \(P_2\) have been received, (5) the users recover from memory their own copies of each others Contact Password \(ConPW_{u_1}^{a_2}\) and \(ConPW_{u_2}^{a_1}\) and they (6) calculate \(\bar{P}_1\) and \(\bar{P}_2\). Lastly, each others identity is authenticated if (7) \(P_1 = \bar{P}_1\) and \(P_2 = \bar{P}_2\).

Similarly to encounters with peers, users also log their encounters with Ad-Dealers. Ad-Dealers advertise their presence in an area by broadcasting messages which are known as Ad-Dealer Location Tags or ADLTs. ADLTs are signed with the Ad-Dealer’s Signing Key \(ADK^{sig}\) and can therefore be verified with the matching verification key \(ADK^{ver}\) which is installed in the user’s client. ADLTs are periodically updated as they contain the Ad-Dealer’s identity \(Aid\) as well as a time-stamp. This prevents replay attacks but also serves additional system functions which are detailed in Chapter 6.

4.2.2 Phase 2: Advert Requesting

Phase 2 takes place when the \(AIP_u\) has been marked with ’TRUE’ consumer interests and the user wishes to obtain relevant adverts. The advert requesting process involves two stages. During the first stage, the user composed a series of Advert Request Messages or ARMs. ARMs contain marked advertising interests from the user’s \(AIP_u\) and are transferred in encrypted form to preserve privacy. For the second stage of the process, the user can choose to either forward the ARMs to an Ad-Dealer directly or via one of the available Agents. In order to obtain his/her adverts as fast as possible, the user approximates the time delay of each available route and selects the optimal option. The technical details of the ARM composition and forwarding procedures are provided in the following paragraphs.

\(^1\)Note that in practice, there might be occasions where rapid encounters will be detected as two devices come in and out of range. For this reason, two encounters are considered as separate events only when a certain amount of time passes in between.
ARM (Advert Request Message) Composition: The user begins the composition ARM\(_i\)s by first creating a sequence of one-time keys which are termed Delivery Keys \((DelK^\text{user})_i\). Since Delivery Keys are symmetrical, they can easily be generated with the use of a hash chain. The user only needs to randomly create the first key and then each consecutive key can be generated by hashing the previous key as denoted in Equation \(4.3\).

\[
DelK^\text{user}_i = h(DelK^\text{user}_{i-1})
\]  

\(4.3\)

To compose the actual ARM\(_i\)s, the user first pairs each of the generated Delivery Keys \((DelK^\text{user})_i\) to Advertising Interests Identifies \(I_{id}\)s which have been marked as ‘TRUE’ in the user’s AIP\(_u\). The user then encrypts the ARM\(_i\)s with the Ad-Dealer’s public key \(SysEK\) (which came pre-installed in the client) as shown in Equation \(4.4\). Encrypted ARM\(_i\)s are labeled with a unique identifier which is referred to as the Order Identity or OrderID.
Lastly, the ARMs and the matching Request Keys are stored in memory until they can be transmitted. Note that the composition ARMs does not need to take place in real time. Users can therefore preserve resources by composing ARMs while their devices are idle.

\[ ARM\_{(OrderID)} = E_{SysEK}[I_{id}, ReqK^{user}] \]  

(4.4)

The user can create a different ARM for each of the 'TRUE' advertising interest of the AIP or pack multiple interests into a single ARM. The number of interests which are contained in an ARM has minimal affect on ADS. In Chapter 5 however, we introduce a mechanism which reduces memory overhead and requires that each interest is composed into a different ARM.

**ARM (Advert Request Message) Forwarding:** ADS users have the option to either deliver their own ARMs directly to an Ad-Dealer or to establish an opportunistic connection via one of the available Agents. To establish the opportunistic network, ADS adopts a history based probabilistic approach which is inspired by PRoPHET [94]. Our opportunistic routing method is based on the calculation of two time metrics, EToC\textsubscript{user} (Estimated Time of Collection) and EToD\textsubscript{Agent} (Estimated Time of Delivery). EToC\textsubscript{user} describes the approximated period after which the user is expected to be able to directly collect adverts from an Ad-Dealer and EToD\textsubscript{Agent} represents the approximated period after which the user is expected to have adverts delivered if she uses the services of a particular Agent. Both the EToC\textsubscript{user} and EToD\textsubscript{Agent} are calculated based on the average time periods between past consecutive encounters. In accordance with similar history-based opportunistic routing protocols such as [94, 41, 135, 156, 37] and [78], the average time value represents the prevalent encounter rate between two network nodes and is used to perform estimated predictions of future encounters. By comparing the two metrics, the user is able to form an intelligent opinion about his/her available options and select the most beneficial.

As illustrated in Figure 4.3, the prospective Requester initiates the interaction by (1) expressing to the Agent his/her intent to possibly use his services. Both users then proceed to (2) calculate their respective EToC\textsubscript{Req} and EToC\textsubscript{Agent}. As shown in Equation 4.5, a user's EToC\textsubscript{user} is calculated by first computing the Average Wait between the user’s past encounters with an Ad-Dealer AW\textsubscript{user}^{'AD'} and subtracting RT\textsubscript{user} which represents the
time which has already passed since his/her last encounter. Upon completion, the Agent sends his own $EToC_{Agent}$ to the prospective Requester and awaits a response. If $EToC_{Req} < EToC_{Agent}$, the prospective Requester sends a negative reply and terminates the interaction. Alternatively, the prospective Requester references her logs and computes the Average Wait between consecutive meetings with the Agent $AW^{Req}_{Agent}$. $AW^{Req}_{Agent}$ represents the expected time intervals between future meetings of the two users. This allows the prospective Requester to calculate the Agent’s $EToD_{Agent}$ as the smallest multiple of $AW^{Req}_{Agent}$ which is greater than $EToC_{Agent}$ or in more simple terms, as the time period until the future meeting which will take place after the Agent has visited the Ad-Dealer. Lastly, if the prospective Requester’s Estimated Time of Collection is smaller than the Agent’s Estimated Time of Delivery ($EToC_{Req} < EToD_{Agent}$), the former sends a negative response or otherwise proceeds to forward ARMs.

$$EToC_{user} = AW^{user}_{AD} - RT_{user}$$ (4.5)

In the example of Figure 4.4, Alice (the prospective Requester) is considering using the services of Bob (an Agent) at time $t_0 = 0$. Alice’s encounters with Ad-Dealers typically take place between intervals of Average Wait $AW^{Alice}_{AD} = 45$ hours and her last encounter was $RT_{Alice} = 24$ hours ago. Alice’s Estimated Time of Collection can therefore be calculated as $EToC_{Alice} = 45 - 24 = 21$ hours which is expected to take place at Collection Time $CT_{Alice} = t_0 + EToC_{Alice} = 21$.

On the other hand, Bob’s Average Wait is $AW^{Bob}_{AD} = 24$ hours and he last encountered an Ad-Dealer $RT_{Bob} = 13$ hours ago. This computes to an Estimated Time of Collection $EToC_{Bob} = 24 - 13 = 11$ hours and places his next encounter at $CT_{Bob} = t_0 + EToC_{Bob} = 11$. Furthermore, Alice and Bob encounter each other between intervals of $AW^{Bob}_{Alice} = 5$ hours which means that their future meetings are expected to take at $t_1 = t_0 + 5, t_2 = t_1 + 5, ..., t_i = t_{i-1} + AW^{Bob}_{Alice}$. Based on this knowledge, Alice can deduce that Bob’s Estimated Time of Delivery $EToD_{Bob}$ is equal to the time of the encounter which takes place immediately after $t_{Bob}$ which means that Delivery Time $DT_{Bob} = t_3 = 15$.

### 4.2.3 Phase 3: Advert Collection

When appearing within vicinity of an Ad-Dealer, users must first authenticate the Ad-Dealer’s identity and then forwards their stored ARMs (if the

---

2Note that when a user’s next encounter with Ad-Dealer is overdue, $EToC_{user}$ returns a negative value.
Figure 4.3: Visual representation of the ARM (Ad-Request Message) forwarding sub-protocol.
user also operates as Agent, this includes ARMs that they ferry on behalf of Requesters). The Ad-Dealer on his part, recovers the Advertising Interests Identifiers $I_{id}$s and Delivery Keys $DelK^user_i$ from the ARMs and uses them to respond with encrypted massages which contain the relevant adverts. The exact technical aspects of the advert collection process are described in the following paragraphs.

**Ad-Dealer Authentication**

Before placing a request, users need to first authenticate the Ad-Dealer. As mentioned before, Ad-Dealers advertise their presence by broadcasting Ad-Dealer Location Tags or ADLTs. ADLTs consist of fourteen integer numbers where the first four indicate the Ad-Dealer’s unique reference number $Aid$, the six following numbers represent the current date in standard format ($DD – MM – YY$) and the last four represent the current time is 24-hour clock format ($hh – mm$). ADLTs are signed with the Ad-Dealer’s Signing
Key $ADK^{sig}$ and can therefore be authenticated with the matching verification key $ADK^{ver}$ which users hold on their clients. Alternatively, more cautious users are given the option for a more secure authenticating by generating their own message (a random nonce) and asking the Ad-Dealer to sign it. In contrast to $ADLT$s which can be pre-computed, signing a user generated nonce takes place in real time and is more taxing for the Ad-Dealer. Ad-Dealers can therefore serve a limited number of authentication requests at the time. In order to protect the Ad-Dealers from being tricked into signing fake $ADLT$s, user-selected messages are not permitted to resemble the format of an $ADLT$. It is therefore a requirement that any random nonce which is offered for signing must include both numbers and letters and have a smaller size than an $ADLT$.

**Request Placement**

Once the Ad-Dealer’s identity has been authenticated (and the encounter has also been logged), the user goes on to forward $ARM$s. The $ARM$s are decrypted by the Ad-Dealer with his private decryption key $SysDK$ and the contained $I_{id}s$ and $DelK^{user}$ are recovered. The Ad-Dealer then forwards the $I_{id}s$ to the Broker and awaits a response.

In order to minimize workload for Ad-Dealers, the Broker stores adverts in encrypted format. As a means of increasing privacy, the Broker can also keep multiple copies of the same advert each encrypted with a different key. The Broker who receives the Ad-Dealer’s $I_{id}s$, replies with the relevant pre-encrypted adverts $AD_1, AD_2, ..., AD_i$ as well as the matching symmetric keys $Key_1, Key_2, ..., Key_i$ which are needed to decrypt the adverts.

As soon as the Broker’s replay is received, the Ad-Dealer organizes the adverts into groups which are termed Bundles. The adverts within a Bundle are identified by a sequence number $SN_i$. The Ad-Dealer then labels each Bundle with a unique reference number which is known as BundleID and finally calculates the hash digest $h(B_i)$ of each Bundle. The Ad-Dealer then constructs Delivery Messages or DMs for each of the received $ARM$s. The DMs are intended as responses to the $ARM$s and contain the information that the user needs in order to locate his/her adverts within a Bundle, verify their integrity and lastly to decrypt them. More specifically each DM contains (1) the BundleIDs of the Bundles where the user’s adverts are stored (2) the hash digest $h(B_i)$ of that particular Bundle, (3) the Sequence Number $SN_i$ of each of the requested adverts within the particular Bundle and (4) the appropriate keys $Key_i$ which are needed to decrypt each advert.

To complete the process, the Ad-Dealer encryps the DMs with the ap-
propriate Delivery Key $DelK^\text{user}_{i}$ (which was included in the ARM) and labels them with the same $OrderID$ ($Order Identity$) as the corresponding ARM as depicted in Equation 4.6. Lastly, the Ad-Dealer sends the Bundles and the encrypted $DM$s back to the submitting user.

$$DM_{(OrderID)} = E_{ReqK^\text{user}_{i}}[BundleID, h(B_i), SN_i, Key_i] \quad (4.6)$$

4.2.4 Phase 4: Advert Delivery

In order to view adverts which have been collected from an Ad-Dealer, the user must first decrypt the received $DM$ with his/her Delivery Key $DelK^\text{user}_{i}$. From within the now decrypted $DM$, the user obtains the appropriate $BundleIDs$ of the Bundles in which his/her adverts are stored, the hash digest $h(B_i)$ of that Bundle, the $Sequence Number$ $SN_i$ of each individual requested advert and the corresponding $Key_i$ which can be used to decrypt each particular advert. With this information, the user is able to locate his/her adverts, verify their integrity and decrypt them.

For $DM$s and adverts which are not addressed to the submitting user but one of his/her Requesters, they are stored in encrypted form until the can be delivered. When the user (Agent) who made the collection from the Ad-Dealer appears within proximity of the Requester, the stored $DM$s can be delivered. After a mutual authentication has taken place, the Requester makes an inquires about the state of his/her order to which the Agent responds with the $DM$s that are addressed to the particular Requester (or with a negative reply if the collection has not yet been made). The Requester decrypts the $DM$s and recovers the $BundleIDs$ of the Bundles in which his/her adverts are stored. The Requester then asks for the specific Bundles and verifies their integrity with the $h(B_i)$. The Requester uses the the $Sequence Number$ $SN_i$ to locate his/her adverts within the set Bundles and finally decrypts them with the appropriate keys $Key_i$. The remaining adverts which are contained in the Bundles but are still encrypted are of no use for the Requester and can therefore be discarded.

Note that if a specific advert has been requested by both the Agent and the Requester, then only a single copy of the advert needs to be sent. This reduces memory overhead but also presents a potential privacy hazard. To ensure privacy, the Ad-Dealer constructs the Bundles in such a way that it is not easily feasible for a curious Agent to guess which adverts are intended for multiple users. This is achieved by placing the adverts with multiple recipients in Bundles which also contain adverts that are individually addressed to the set users. Naturally, this approach consumes additional bandwidth.
since the Requester needs to download data which is later discarded but at the same time preserves his/her privacy against the Agent.

4.3 Protocol

The ADS protocol is depicted in Figure 4.5 is used for the acquisition of adverts.

1. The user computes a symmetric key Del$K_U$. The user then composes $ARM_U$ (Advert Request Message) which contains Del$K_U$ and the marked interests $I_{id}$ from his/her Advertising Interest Profile $AIP_u$. Lastly, the $ARM_U$ is encrypted with the System Encryption Key Sys$EK$.

2. The $ARM_U$ is sent to an Agent.

3. The Agent forwards the user’s $ARM_U$ along with his/her own $ARM_A$ to an Ad-Dealer.

4. The Ad-Dealer decrypts both $ARM$s with the corresponding System Decryption Key Sys$DK$.

5. The Ad-Dealer forwards the contained interests $I_{id}$ to the Broker.

6. Upon receiving the Ad-Dealer’s message, the Broker recovers the appropriate adverts in encrypted form.

7. The Broker sends the encrypted adverts and matching keys back to the Ad-Dealer.

8. The Ad-Dealer organizes the adverts into Bundles and composes the keys into Delivery Messages $DM_U$ and $DM_A$ which are encrypted with the keys Del$K_U$ and Del$K_A$ that he received from the $ARM$s.

9. The $DM$s and the Bundles are sent back to the requesting Agent.

10. The Agent decrypts $DM_A$ and uses the contained information to locate and decrypt his/her adverts from within the Bundle.

11. The Agent forwards $DM_U$ and the Bundle to the user.

12. The user decrypts $DM_A$ and uses the contained information to locate and decrypt his/her adverts from within the appropriate Bundle. The remaining adverts, which are still encrypted, are discarded.
4.4 Evaluation

In the section we evaluate \textit{ADS} in accordance with the system requirements which we determined in Section 4.1.3. The primary focus of our evaluation is privacy and security for all system stakeholders. The practicality and efficacy of \textit{ADS} are further enhanced by supplementary sub-systems which are evaluated separately in a later chapter.

4.4.1 User Privacy Against Ad-Dealers

By incorporating opportunistic networking to anonymous-download systems, \textit{ADS} not only extends the system’s reach but also offers an increased level of privacy compared to contemporary designs. Intermediate nodes of
the opportunistic network \( (\text{Agents}) \) function as proxies which further mask the \textit{Requester}'s identity from \textit{Ad-Dealers}. An inherit benefit of this setup lies on the fact that \textit{Agents} are themselves users and therefore have no immediate benefit from colluding with \textit{Ad-Dealers} to compromise \textit{Requesters}, especially those who belong in the same social cycle. In addition to this, the availability of different \textit{Agents} distributes risk of compromise in contrast to most proxy architectures which present a single point of failure. A side benefit of the opportunistic network is that it also offers additional security even for users who directly connect to \textit{Ad-Dealers}. Since the system allows for \textit{ARMs} (\textit{Advert Request Messages}) from multiple users to be aggregated and delivered by a single \textit{Agent}, an \textit{Ad-Dealer} is not able to distinguish if a specific request belongs to the submitting user or if set user operates as \textit{Agent} on behalf of a \textit{Requester}. However, it needs to be stated that \textit{Ad-Dealers} may potentially be able to bypass this security measure by analyzing the keys which are included within the \textit{ARMs}. As users generate encryption keys with the use of a hush-chain, \textit{Ad-Dealers} are able to identify subsequent \textit{ARMs} that were created by the same user even if different \textit{Agents} are used for the transmission of each \textit{ARM}. It is therefore necessary for users to update their keys as often as possible. From a practical point of view, this is entirely feasible as the system uses symmetric keys which can be created with relative ease.

One last aspect that needs to be considered is how the opportunistic network could potentially have negative consequences should user anonymity were to be compromised. If \textit{Ad-Dealers} were to uncover the identities of users, then not only would they be able to associate them to their advertising interests but they would also learn of their social affiliations with each other. However, the only way that user anonymity could be exposed would be if the \textit{Broker} were to tamper with the mobile clients which could easily be detected. Despite not being easily feasible to achieve without being exposed, such behavior constitutes as malicious and therefore falls outside the scope of this research which assumes the \textit{Broker} as curious but honest.

### 4.4.2 User Privacy Against Curious Users

\textit{ADS} ensures user privacy against a curious user \( \tilde{U} \) by enforcing the use of cryptography. \textit{ARMs} (\textit{Advert Request Messages}) are encrypted with the \textit{System Encryption Key} \( \text{SysEK} \) and can therefore not be decrypted by \( \tilde{U} \) as he/she does not have access to the corresponding \textit{System Decryption Key} \( \text{SysDK} \) which remains private among \textit{Ad-Dealers}. By analyzing the size of encrypted \textit{ARMs}, \( \tilde{U} \) may infer the number of contained \textit{Advert Interests}.
CHAPTER 4. ADS: ADVERT DISTRIBUTION SYSTEM

Identities $I_{ids}$ however, $\bar{U}$ is not able to uncover which specific $I_{ids}$ by launching a known plain-text attack (attempting to identify the cryptograms by encrypting her own ARMs). Since ARMs consist of a user selected Delivery Key $DelK_{i}^{user}$, it is unlikely that any two ARMs will produce the same cryptograms even if they happen to contain the exact same $I_{ids}$.

Adverts are also encrypted with Broker generated keys $Key_{i}$ which are contained within DMs (Delivery Messages). DMs are themselves encrypted with Delivery Keys $DelK_{i}^{user}$ and are therefore visible only to the user who constructed the matching ARMs. Adverts which are addressed to more than one users are always sent in Bundles which also consist of adverts which are individually addressed to the respective users. $\bar{U}$ can therefore not access adverts that are not addressed to him/her nor identify which adverts are addressed to multiple recipients. However, it is possible for $\bar{U}$ to uncover the interests of other users without having to decrypt DMs. If $\bar{U}$ were to place an order for all adverts which are available in the system, he/she would have access to every $Key_{i}$. $\bar{U}$ could then operate as Agent and determine the interests of her Requesters by simply decrypting all the adverts which are included within each Bundle. To prevent this attack from being effective, the Broker makes sure to regularly update the keys $Key_{i}$ which are used for the encryption of adverts. The Broker is able to do that with minimal overhead since $Key_{i}$ are symmetric keys and therefore relatively easy to generate.

4.4.3 User Security Against Malicious Users

The spread of harmful content in opportunistic networks is a serious threat that has not been addressed by analogous advert distribution systems. As data is propagated through peer-to-peer connections, fake adverts could quirky spread across the network and infect multiple nodes before being detected. In opportunistic connections with multiple intermediate nodes, it would also be very difficult to detect and isolate the perpetrator of the attack, especially while still maintaining user privacy.

$ADS$ minimizes the threat of fake adverts by incorporating cryptographic countermeasures which enable users to easily identify content which has not been dispatched by valid Ad-Dealers. Ad-Dealers dispense Bundles of encrypted adverts and DMs to Agents with the intent for them to be delivered to their Requesters. In order for a malicious Agent $\bar{U}$ to replace or insert a fake advert $\bar{AD}_{i}$ into a Bundle, he/she would need to first encrypt $\bar{AD}_{i}$ with the appropriate key $Key_{i}$. This presents us with two possible attacks scenarios as detailed in the following paragraphs.

If $\bar{U}$ has no access to $Key_{i}$, he/she has no other option but to use a
different $\hat{\text{Key}}_i$ of his/her own choosing. For this attack to be effective, $\hat{U}$ would also need to replace the original $\text{Key}_i$ with $\hat{\text{Key}}_i$ within the $\text{DMs}$ which are addressed to the Requester. The Requester’s $\text{DMs}$ are however encrypted with a Delivery Key $\text{DelK}^\text{Requester}_i$ which $\hat{U}$ does not know. For $\hat{U}$ to gain access to $\text{DelK}^\text{Requester}_i$, he/she would first need to decrypt the Requester’s $\text{ARMs}$ which is not possible without the System Decryption Key $\text{SysDK}$ that remains private among Ad-Dealers. If $\hat{U}$ were to only replace the original advert $\text{AD}_i$ with the fake advert $\hat{\text{AD}}_i$ without updating the $\text{DMs}$, his/her attack would fail since the victimized Requester would not be able to decrypt $\hat{\text{AD}}_i$ with the original key $\text{Key}_i$ which was obtained form the received $\text{DM}$.

$\hat{U}$ can potentially obtain $\text{Key}_i$ by also submitting a request for the particular $\text{AD}_i$. Since $\text{Key}_i$ is a systematic key, $\hat{U}$ could use it to encrypt a fake advert $\hat{\text{AD}}_i$ and replace it in the Biddle without the need to alter the Requester’s $\text{DMs}$. The victimized Requester would then be able to decrypt $\hat{\text{AD}}_i$ with the original $\text{Key}_i$ which was obtained by decryption his/her $\text{DMs}$. To prevent such an attack, $\text{DMs}$ include a hush digest $h(B_i)$ of each Bundle that contains adverts which are addressed to the requesting user. The Requester can use $h(B_i)$ to verify the integrity of a Bundle after receiving them from $\hat{U}$. If the confirmation were to fail, the Requester would had been able to perceive the attack without even having to decrypt any adverts.

### 4.4.4 Robustness Against Sabotage Attacks

Contemporary advertising systems often ignore the threat of sabotage. Even when user privacy is protected, an attacker can still undermine the integrity of a system by interfering with its correct operation. In the case of advertising systems (especially those which make use of decentralized architectures), sabotage comes in the form of impersonation attacks or spoofing. To launch a spoofing attack, an attacker impersonates the identify of a legitimate system stakeholder in order to trick a victim into performing a certain operation. The stakeholders of $\text{ADS}$ who may be threatened with impersonation attacks are users and Ad-Dealers. The advent of an impersonation attack presents us with four potential scenarios as detailed in the following paragraphs.

**Attack Scenario 1:** For our first scenario, the identity of a user $U$ is spoofed by an attacker $\hat{U}$ in order to victimize a second user $U_{\text{vic}}$. If $U$ and $U_{\text{vic}}$ are registered Contacts, $\hat{U}$ can trick $U_{\text{vic}}$ into logging fake encounters with $U$ which will decrease the Average Wait between consecutive meetings...
This will interfere with the correct operation of the opportunistic routing algorithm as $U_{vic}$ can be deceived into selecting $U$ or $\hat{U}$ as his/her Agent. This will result in $U_{vic}$ either revising adverts slower that originally anticipated or even not receiving them at all.

**Attack Scenario 2:** For the second scenario, the identity of Agent $A$ is impersonated by an attacker $\hat{A}$ in order to victimize one or the Agent’s Requesters $R$. During an interaction with $\hat{A}$, $R$ could be tricked into downloading fake adverts. Although ADS protects $R$ from falling victim to fake adverts, $R$ would still be deceived into believing that the real Agent $A$ is malicious and refrain from using his/her services in the future.

**Attack Scenario 3:** For the third scenario, the identity of a Requester $R$ is impersonated by an attacker $\hat{R}$ in order to victimize an Agent $A$. The encounter between $\hat{R}$ and $A$ takes place before $A$ has had the chance to make a collection from an Ad-Dealer, then the effects of the attack will be minimal as $\hat{R}$ will make an inquiry about the state of his/her order to which $A$ will simple respond negatively. However, if the encounter takes place after $A$ has visited an Ad-Dealer and before he/she had the chance to meet the real $R$, then $A$ will be tricked into forwarding to $\hat{R}$ the DMs which are addressed to $R$. The privacy of $R$ is not compromised as DMs are encrypted, however $A$ will be deceived into thinking that a successful delivery has taken place and will discard the DMs which will prevent him/her from completing the actual delivery when the meeting with the real $R$ takes place.

**Attack Scenario 4:** For the final scenario, an attacker $\hat{D}$ impersonates the identity of an Ad-Dealer to victimize a user $U$ who appears within proximity. This can result in $U$ sending ARM$s$ to $\hat{D}$ and possibly downloading fake adverts and DM$s$. The effects of this attack would be minimal as only registered Ad-Dealers are capable of decrypting ARM$s$ and encrypting adverts or DM$s$. However, if $U$ is not interested in making a collection at that time, he/she will still log a fake encounter with an Ad-Dealer which will effect his/her Average Wait $AW_U^{\text{AD}}$ and potentially interfere with the operation of the opportunistic network.

**Attack Prevention:** To counter the aforementioned attacks, ADS incorporates strong authentication mechanisms. Users are authenticated with mutual password verification which prevents them from logging encounters
and exchanging data anyone other than their registered contacts. The protocol protects against reply attacks by incorporating a standard challenge-response handshake and prevents attackers from stealing or guessing passwords by using out-of-band channels and cryptographically secure hush functions. Ad-Dealers are authenticated by broadcasting data which have been singed with the Ad-Dealer’s Signing Key $ADK^{sig}$ and can therefore be verified by users with $ADK^{ver}$. The signed data are either Ad-Dealer Location Tags ADLTs which contain time-stamps or random messages which are selected by the users themselves. The Ad-Dealer’s Signing Key $ADK^{sig}$ is only used for the verification and Ad-Dealers are prevented from signing fake ADLTs as user-selected messages which are requested for signing are not permitted to resemble the standard format of ADLTs. The signing of ADLTs and random messages prevents replay attacks but is still susceptible to Man-in-the-Middle attacks. To perform a Man-in-the-Middle attack, an adversary needs to collect ADLTs from one location and rebroadcast them at a different location before the time-stamps are expired. The damage from such an attack is limited as the victim is tricked into logging fake encounters with an Ad-Dealer but user privacy is still not compromised. Man-in-the-Middle attacks are difficult to protect against but are also very impractical and therefore pose minimal risk for the system. Overall, the integrity of ADS is substantially protected against all conceivable forms of sabotage.

4.5 ADS: Advert Distribution System Summary

In this chapter we presented ADS as a privacy-oriented alternative to the currently adopted OBA system. ADS expands on previous work by combining anonymous-download technologies with opportunistic networking in order to provide a advertising distribution scheme which overcomes the limitations of older designs. The system’s infrastructure is based on a multilayered architecture which is practical, inexpensive and simple to launch as well as to manage. Through a series of quality evaluation scenarios, we demonstrated that ADS offers notable user privacy against all other concerned parties (including other user) and also provides additional security against impersonation attacks and sabotage. Overall, ADS was shown to offer considerable advantages in comparison to contemporary systems in terms of privacy, security and practicality.
Chapter 5

Private Profile Comparison

Resource consumption is an inherent issue of opportunistic networking as nodes are required to download and store data on behalf of their peers. To make matters worse, many opportunistic routing protocols generate multiple copies of the same data which places additional strain on the already limited resources of mobile devices. As demonstrated in Chapter 4 Section 4.2.4, ADS reduces bandwidth and memory consumption by implementing an encryption scheme which allows a single copy of encrypted data to be delivered to multiple mobile users. When an Agent requests the same advert as one of the Requesters that they serve, the Ad-Dealer only needs to send a single advert which is addressed to both users.

Relevant research has shown that users within the same social network tend to share cultural preferences and influence each others behavioural inclinations [91, 112, 93, 31]. It therefore stands to reason that Agents and Requesters within ADS are likely to have shared advertising interests. Considering however that Requesters select which advert to request from each prospective Agent at random, it is understandable that the probability of requesting an advert of shared interest is entirely left to chance. This offers the optimal level of privacy but at the same time fails to exploit the system’s full potential at conserving resources. Alternatively, if Agents and Requesters were to openly compare their respective Advertising Interest Profiles $AIP_A$ and $AIP_R$ in order to identify shared interests, this would result in maximized resource conservation but it would come at the expense of privacy. To strike a trade-off between resource utilization and privacy, we propose a variety of profile comparison mechanisms. The techniques we implement in our designs increase the probability of selecting adverts of shared interest but still maintain an acceptable level of privacy.
5.1 Profile Comparison

A user’s Advertising Interests Profile (AIP) is represented as a list of common advertising interests which are marked as either ‘TRUE’ or ‘FALSE’. The AIP is kept locally where it is updated dynamically based on the user’s activity. By performing a profile comparison, a Requester is able to select a set of ‘TRUE’ interests out of their AIP\(_R\) with an increased probability that the same interests are also marked ‘TRUE’ within the Agent’s AIP\(_A\). The challenging aspect of the profile comparison however, is that it needs to be performed in such a way that user privacy is preserved. This presents us with a paradox as we aim to increase the probability of two users selecting a shared interest but not to the point where the selection of a shared interest is guaranteed. If a profile comparison were to always guarantee the selection of a shared interests, then privacy would be compromised. In Sections 5.1.1 to 5.1.3, we present a variety of profile comparison algorithms which are supported by ADS. In Sections 5.2 we experimentally test the performance of our design via a series of simulations. Finally, in Sections 5.3 we discuss the results of our experiments and evaluate our design.

5.1.1 D-PC: Demographic Profile Comparison

Demographics offer essential insight towards targeting certain groups of consumers with adverts which best fit their needs [117]. Demographic advertising is the practice of segmenting a consumer audience into groups based on demographic attributes. Each particular group is then targeted with adverts which best associate to the characteristics of their demographics (e.g., adverts for luxury designer dresses which target adult females of high income). For this type of advert targeting to be possible, it is necessary for a third party (i.e. the Ad-Network) to be able to classify advert viewers into demographic groups based on attributes such as gender, age, location and income. Inevitably, this creates a significant privacy breach for consumers as it requires the Ad-Network to have access to demographic information which may be considered sensitive.

Demographic Profile Comparison (D-PC) exploits the same principles as demographic advertising for an entirely different purpose. D-PC compares the demographic attributes of two users in order to identify advertising interests which may be relevant to both. One important aspect that needs to be considered at this point is how D-PC affects privacy. In traditional targeting, all consumer demographics are considered sensitive as they expose
private information to an untrustworthy third party (i.e. the Ad-Network). It can however be argued that during face-to-face interactions, demographic attributes which are already known about each user cannot be considered as sensitive. Based on this notion, we can identify two types of demographic attributes which are analyzed in Subsection 5.1.1.

**Demographic Attribute Classification**

The demographic attributes which are exploited by D-PC can be classified into two types, (1) physical and (2) social. Physical attributes are traits that two users can easily infer about each other through simple observation. Social attributes are traits that one user can deduce about the other based on their social interactions. Considering the fact that each pair of users have a different set of social interactions, it is evident that social attributes of a user are conditional to the second user who makes the assessment. For example, the social attributes which are assigned to Alice by Bob may be entirely different to the social attributes which are assigned to Alice by Charlie. The details of the exact manner in which social attributes may differ between sets of users are made clear in the following paragraphs.

As shown in Figure 5.1, the physical attributes which are exploited by D-PC are Gender and Age and the social attributes are Location and Status. For the purposes of our system, we assume Gender as binary which can be classified as either 'Male' or 'Female' while Age takes a discrete value that represents particular age groups which are typically used in advertising. These are listed in Figure 5.1 as '< 18' for minors, 'Young Adult' for the ages between 18 and 34, 'Adult' for 35 to 50 and '> 50' for users who are older than 50. Gender and Age are manually selected by all users as a precondition to use the service. Users who assume the role of Agent publicize their physical attributes to their peers (prospective Requesters) when they register as Contacts. This is done for reasons of simplicity as it would be impractical if it was the Requester who had to manually insert the gender and age of each Agent that they came into contact with. For an Agent to publish false information would be counter intuitive as the entire premise of a profile comparison is to reduce overhead on the Agent’s side. Even if an Agent were to act against their interest, this would have no effect on the Requester.

As previously stated, the physical attributes of a particular user remain constant while social attributes differ based on the second user who makes the observation. It is therefore possible for an Agent to have different social attributes for different Requesters. Social attributes can be deduced by a
CHAPTER 5. PRIVATE PROFILE COMPARISON

Requester after socializing with an Agent. The two social attributes that are administered by D-PC are Location and Status. Location is an attribute which expresses the physical locations where two users typically encounter each other. ADS is able to determine the Location attribute by processing data that it already has in its disposal. More specifically, in Chapter 4 Section 4.2.1 we explained that ADS tracks the user’s visited locations and also tracks the user’s encounters with her peers. We briefly remind you that location tracking is used to determine the user’s advertising needs while encounters with peers are logged for the purpose of establishing opportunistic connections. D-PC takes advantage of encounter logs and combines it with location tracking in order to pinpoint the locations where user sightings typically happen. In more detail, when Alice logs a meeting with Bob, she also registers the location of the meeting (when that is available). After a few meetings have taken place, Alice can refer back to her records and derive in which locations she and Bob meet the most often. Based on this knowledge, Alice can infer that she and Bob are likely to have a mutual interest for adverts which are relevant to those particular locations.

For reasons of both practicality and security, visited locations are not kept as GPS coordinates. For our needs, it is only necessary to keep the type of location in terms of its social utility (e.g., home, office, restaurant). D-PC adopts the same taxonomy as Foursquare where venues are classified in categories and subcategories based on their utility. Figure 5.1 lists some of the main categories of venues which are ‘Residence’, ‘Work related’, ‘Education’, ‘Nightlife’, ‘Entertainment’, ‘Food related’, ‘Recreation’ and ‘Travel’. These categories can be further broken down in more specific subcategories. For example, the venue category ‘Entertainment’ includes a subcategory ‘Performing Arts’ which contains ‘Opera’, ‘Theater’ and ‘Dance Studio’. For each of the available venue categories and subcategories, meeting locations are logged as a series of counters with each of them representing either a category or subcategory of venue. Depending on the information that is available, the user raises the appropriate counter or counters. For example, if a meeting is registered inside an office, the device will raise the counters for both the general category which is ‘Work-related’ as well as for the specific subcategory which is ‘office’. However, if the exact type of facility is not known, the device will only raise the counter for ‘work-related’.

The last social attribute which is used by D-PC is labeled in Figure 5.1 as ‘Status’. Unlike the previously explored attributes of Gender, Age and Location which are common in advertising systems, Status is unique to D-PC and is used to classify the social relationship of a pair of users. The three possible Status classifications are ‘Close’, ‘Casual’ and Professional’. These
classifications are obtained based on user meeting patterns in terms of time, duration and location. More specifically, when two users regularly appear within proximity in residential locations for extended periods of time, they can be classified as having a 'Close' Status. This classification indicates a very strong social relation such as that of spouses, family members and close friends. Pairs of users who have more irregular meetings that mostly take place in locations which relate to entertainment, nightlife or recreation can be classified as having a 'Casual' Status. This classification is expected from people who relate as friends and acquaintances. Finally, users who meet on a daily basis during work hours and within locations which relate to work or education can be classified as having a 'Professional' Status. Such classifications is expected of coworkers, colleagues and associates. One final detail that needs to be stated is that the 'Status' field is left blank when the available data is too ambiguous to yield a definitive classification. The absence of a Status attribute will have limited impact on the overall performance as Status only serves as a supplementary piece of information.

Figure 5.1: Demographic attributes which was supported by D-PC (Demographic Profile Comparison).
CHAPTER 5. PRIVATE PROFILE COMPARISON

Interest Selection

Similarly to conventional targeting, D-PC requires that the advertising interests which can be found in the AIP be associated to specific demographic groups. To better comprehend this notion, consider a simple example where we apply a single demographic attribute such as gender. The entries of the AIP which target a specific gender can be classified as 'Male' or 'Female' and all the remaining entries which are relevant to either gender are classified as Neutral'. When a D-PC between two users of different genders takes place, gender-specific interests are excluded and the focus of the selection can be limited to interests which are labeled as 'Neutral'. Similarly, when two male users perform a D-PC, female-specific interests are excluded and the selection is focused on the remaining interests which are classified as 'Male' and Neutral'.

The association of interests to consumer demographics can be achieved with the same market research models which are currently in use for traditional advertising. Gender, Age, and Location are all demographic attributes which are very commonly used in such advertising models and can therefore be easily adapted to work with D-PC. Apart from the three aforementioned demographics, market models also use other attributes such as occupation, educational level, marital status, religion, and income. The aforementioned demographics were deliberately excluded from D-PC due to their sensitive nature even in face-to-face interactions among peers. However, we need to clarify at this point that the sensitivity of the same demographics is not always absolute but may be conditional to the social relation that two users share. For example, Alice may be reluctant to reveal her occupation to Bob but her occupation is already known to Charlie who is her coworker. The Status attribute is used by D-PC to specifically exploit this provisional form of sensitive information. Status indicates the social relation between a pair of users which in turn allows D-PC to infer certain characteristics that these users may have in common. More specifically, a 'Close' Status which is attributed to spouses and family members may reveal that the two users are likely to have a shared ethnicity, religion, and socioeconomic status [82]. Users who have a friendly relationship and are therefore classified with a 'Casual' Status, may share the same interests in terms of social lifestyle (for e.g., hobbies, recreational activities, etc.) [142, 143]. Finally, users who have a 'Professional' Status can be assumed to have the same occupation, educational level, and social class [102]. As is the case in traditional demographic targeting, all of the affirmations demographics can be associated to specific advertising interests.
To perform the D-PC, the Requester begins by excluding all the interests that do not comply with the Agent’s physical demographic attributes of Gender and Age. The Requester then selects of the remaining interests the ones which best relate to the social attributes that they share with the Agent. For example, Bob, Charlie and Dave are all male adults who work for a tech company. Most of their meetings take place in company ground during work hours and are therefore classified (in pairs) as having a ’Professional’ Status. However, Bob and Dave are also seen as having meetings outside the workplace in venues which associate to recreation (e.g., gym, park, swimming pool). Based on this knowledge, an D-PC between Bob and Charlie is likely to yield a interests which relates to their occupation such as a personal electronic device (e.g., a new smart-phone) but an D-PC between Bob and Dave is more likely to result in an interest which also relates to athletic lifestyle (e.g., a wearable activity monitor). The exact demographic model which is used for the selection of interests can operate independently to the rest of the system. This offers great flexibility as it allows for the adoption of existing advertising designs or the development of entirely new ones.

5.1.2 F-PC: Fragmented Profile Comparison

*Fragmented Profile Comparison (F-PC)* follows a probabilistic approach which calls for the separation of the *Agent’s Advert Interest Profiles AIP_A* into *f* fragments which are then assigned a ranking score based on the total number of marked interests they contain. The rankings are shared with the prospective Requester who is then able to increase her chances of selecting a common interest by focusing on the highest ranked fragments. In order for the ranking scores to accurately represent the probabilities of selecting a marked interest, the fragments need to be of equal size which requires that the number of fragments *f* is a divisor of *n* (where *n* is the number of total interests within the standard AIP format). However, such a restriction on the possible values of *f* significantly limits the system. To therefore allow for more flexibility, the system can also accept any *f* which results in the last fragment to be slightly larger or smaller within a certain margin. For example, an *AIP* or size *n = 100* may be separated into *f = 3* fragments of sizes *F_1 = 33*, *F_2 = 33* and *F_3 = 34*.

The *Agent* initiates the F-PC by selecting *f* and sending it to the Requester. When a large *f* is selected, the size of fragments becomes smaller which increases the effectiveness of the comparison but at the same time reduces the *Agent’s* privacy. In contrast, a small *f* of bigger size fragments
offers more privacy for the Agent but comes at the expense of greater overhead as it results in the selection of fewer common interests. It falls within the discretion of the Agent to select an appropriate \( f \) which best serves their priorities between privacy and overhead. Based on the selected \( f \), the two users separate their respective profiles \( AIP_A \) and \( AIP_R \) into fragments which are identified by a sequence number \( F_i \) (e.g., \( F_1, F_2, ..., F_f \)). The fragments of the Agent's \( AIP_A \) are then ranked based on the number of interests that are marked 'TRUE' in each one. The highest ranking is assigned to the fragment with the most marked interests while the lowest ranking is reserved for the fragment with the least marked interests. The Agent then composes the fragment identifiers \( F_i \) in descending order into a list \( L \) which is shared with the Requester. Note that \( L \) contains only the fragment identifiers \( F_i \) and not the fragments themselves. The Requester does not therefore learn which of the Agent's interests are marked as 'TRUE' nor the number of marked interests in each fragment. The only piece of information which is obtained by Requester is the order in which the Agent's fragments are ranked (i.e. which fragments contain the most interests). This allows the Requester to have a general overview of the Agent's \( AIP_A \) but still preserves the Agent's privacy.

Upon receiving the ranked list \( L \), the Requester proceeds to select interests out of her own \( AIP_R \) by prioritizing on entries out of fragments that have the highest ranking in \( L \). By focusing her selection on the highest ranking fragments, the Requester increases her chance of selecting a shared interest while still remaining unaware of the marked interests within the Agent's \( AIP_A \). After the selection has taken place, the Requester is free to construct an ARM (Advert Request Message) with the selected interests and forward it to the Agent.

In the example which is presented in Figure 5.2, the Agent's \( AIP_A \) of size \( n = 200 \) is separated in \( f = 5 \) fragments of 40 interests. The fragments are ranked and the fragment identifiers \( F_1 \) to \( F_5 \) are placed in descending order in a list \( L = \{F_5, F_1, F_4, F_2, F_3\} \) which is then made public to the Requester. Upon receiving \( L \), the Requester can select marked interests out of her own \( AIP_R \) by focusing their selection on higher ranked fragments such as \( F_5 \) or \( F_1 \). Gaining access to \( L \) allows the Requester to know which fragment of the Agent's \( AIP_A \) contains the most marked interests but she has no way of inferring which interests or even how many interests are marked.
5.1.3 S-PC: Selective Profile Comparison

Selective Profile Comparison (S-PC) is based on the principle that a Requester is given the ability to perform multiple Candidate Selections (CSs) (e.g., CS$_1$, CS$_2$, . . ., CS$_i$) of interests and propose them to the Agent. The Agent can then compare the proposed CSs and accept the one which offers the best conservation of bandwidth. To perform the selection, the Agent needs to be able to calculate the number of shared interests in each CS. However, the Agent should not be able to see which exact interests are included in each CS as this would compromise the Requester’s privacy. Performing the aforementioned selection is made possible through the use of homomorphic encryption.

Homomorphic encryption is a cryptographic method which allows for the calculation of certain mathematical operations on encrypted data. Such cryptographic schemes have gained considerable notoriety in recent years and especially in the research field of cloud computing. The Paillier [114]
public key crypto-system is such a homomorphic scheme which exhibits an additive property. More specifically, given two messages \( m_1 \) and \( m_2 \) which produce cipher-texts \( c_1 = E\{m_1\} \) and \( c_2 = E\{m_2\} \), we are able to calculate the cipher-text of their sum \( c_3 = E\{(m_1 + m_2)\} \) from the multiplication \( c_3 = c_1 \ast c_2 \). This enables an non-trusted party to compute \( E\{(m_1 + m_2)\} \) without knowing \( m_1 \) and \( m_2 \). Additionally, Paillier also has a self-blinding property that allows for one cipher-text to be changed into another without effecting decryption. Given \( c = E\{m\} \) we can therefore compute \( c' \) so that \( D\{c'\} = m \).

The Agent initiates the S-PC by computing a pair of asymmetric encryption keys. The decryption key is stored for future use and the encryption key is used to produce an encrypted version of the Agent’s User Interest Profile \( \hat{AIP}_A = E\{AIP_A\} \). Note that each of the entries of \( \hat{AIP}_A \) (‘ones’ and ‘zeros’) are encrypted as individual digits and not as a binary number. For an \( AIP_A = [I_1 \ I_2 \ \cdots \ I_x] \) we should have \( \hat{AIP}_A = [C_1 \ C_2 \ \cdots \ C_x] \).

It needs to be stated that two equal interests \( I_x = I_y \) are going to produce cipher-texts that are not equal to each other \( C_x \neq C_y \). This is one of the features of Paillier which ensures that no two cryptograms are the same even if they are produced from the same plain-text. To achieve this, Paillier incorporates into the encryption method a random input which effects the form of the cipher-text but does not effect the way the cipher-text is decrypted.

A detailed explanation on the inner workings of this process with numerical examples are demonstrated by Sridokmai et al. [130]. After the encryption has been performed, the generated \( \hat{AIP}_A \) is sent to the Requester but the decryption key remains with the Agent. The Requester is therefore not able to see if a particular interest is marked as ‘TRUE’ or ‘FALSE’ in \( \hat{AIP}_A \).

Upon receiving \( \hat{AIP}_A \), the Requester performs a series of CSs (Candidate Selections) and then precedes to calculate their weights \( W_{CS} \). The calculation of the weight for a CS can be performed by multiplying the specific cipher-texts of \( \hat{AIP}_A \) which correspond to the particular interests which are contained within \( CS \). For example, for \( CS_j = [I_x \ I_y \ I_z] \) we have \( W_{CS_j} = (C_x \ast C_y \ast C_z) \). Considering the additive property of Paillier, we are able to infer that \( W_{CS_j} = E\{I_x + I_y + I_z\} \) and since the value of each \( I \) is either ‘one’ or ‘zero’, it also holds true that \( D\{W_{CS_j}\} \) is equal to the number of interests in \( CS_j \) which were marked as ‘TRUE’ in \( AIP_A \). Before sending the produced weights \( (W_{CS_1}, W_{CS_2}, \ldots, W_{CS_i}) \) back to the Agent however, the Requester must first alter them by exploiting the self-blinding property. This is done as a means of enhancing security as otherwise the Agent would have been able to deduce the three cipher-texts which produce
a particular wight $W_{CS_j}$ by exhausting all possible combinations. Lastly, when the Agent receives the produced weights, he/she decrypts them, selects the CS with the highest wight $W_{CS}$ and notifies the Requester so that the appropriate ARMs may be forwarded.

To better comprehend the function of S-PC, consider the following simple example. The Agent sends to the Requester an encrypted profile of ten interests $\overline{AIP_A} = [C_1 \ C_2 \ C_3 \ C_4 \ C_5 \ C_6 \ C_7 \ C_8 \ C_9 \ C_{10}]$ which is generated from a corresponding $AIP_A = [1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0] \ (I_1 = I_3 = I_5 = I_7 = I_9 = 1 \ and \ I_2 = I_4 = I_6 = I_8 = I_{10} = 0)$. The Requester makes two $CS$s of three interests $CS_1$ and $CS_2$ and computes their wights as $W_{CS_1} = (C_1 \ast C_3 \ast C_5)$ and $W_{CS_2} = (C_6 \ast C_8 \ast C_{10})$. When the additive property of Paillier is considered, we can infer that $W_{CS_1} = E\{I_1 + I_3 + I_5\} = E\{1 + 1 + 1\} = E\{3\}$ and in similar fashion $W_{CS_2} = E\{I_6 + I_8 + I_{10}\} = E\{0 + 0 + 0\} = E\{0\}$. Lastly, the Requester applies the self-blinding property on $W_{CS_1}$ and $W_{CS_2}$ in order to produce $\overline{W_{CS_1}}$ and $\overline{W_{CS_2}}$ before sending them back to the Agent. By decrypting $\overline{W_{CS_1}}$ and $\overline{W_{CS_2}}$, the Agent receives the weights $W_{CS_1} = 3$ and $W_{CS_2} = 0$. Based on the results, the Agent is able to deduce that $CS_1$ contains three interest which are are common to his/her own while $CS_2$ contains none. However, the Agent remains unaware of the exact contents of $CS_1$ and $CS_2$ as there is no way of knowing which of the encrypted entries in $\overline{AIP_A}$ were added in order to produce $W_{CS_1}$ and $W_{CS_2}$.

**CS-PC: Collaborative-Selective Profile Comparison**

Collaborative-Selective Profile Comparison (CS-PC) is an extension of S-PC which introduces multiple Requesters. When CS-PC is supported, the Agent composes his/her encrypted profile $\overline{AIP_A}$ in the same way as S-PC and sends it to a Requester who is referenced as the Prime Requester $R_0$. To perform an CS-PC, $R_0$ composes a series of Candidate Selections $CS_1^0, CS_2^0, \ldots, CS_9^0$ and calculates the weights $W_{CS_1}^0, W_{CS_2}^0, \ldots, W_{CS_9}^0$. $\overline{AIP_A}$ is then forwarded by $R_0$ to $r$ Secondary Requesters $R_1, R_2, \ldots, R_r$ who compose their own Candidate Selections and calculate the respective weights\(^1\). The Secondary Requesters then forward the weights and a set of matching ARMs for each Candidate Selection back to $R_0$. $R_0$ exploits the additive property of Paillier in order to compute a set of overall weights $OW_{CS_i}$ as show in Equation 5.1 and forwards them to the Agent.

\(^1\)The weights have all gone through the self-blinding operation which prevents $R_0$ from determining the content of the CSs of the Secondary Requesters.
CHAPTER 5. PRIVATE PROFILE COMPARISON

The Agent then performs a selection based on the decrypted weights and informs $R_0$ so that the appropriate ARMs (of both the Prime Requester and the Secondary Requesters) can be forwarded. The remaining ARMs which correspond to the Candidate Selections that were not selected are discarded.

$$O W_{CS_i} = \sum_{k=0}^{r} W^k_{CS_i}$$  \hspace{1cm} (5.1)

To complete the delivery, the Agent collects adverts for all Requesters but only sends them to the Prime Requester $R_0$. Beyond that, it is the responsibility of $R_0$ to track down the Secondary Requesters and forward their adverts. In a sense, the Prime Requester serves the role of a secondary Agent who further extends the reach of the opportunistic network to more users. However, serving the Secondary Requesters also increases the system’s complexity and may possibly result in delayed delivery for the Secondary Requesters.

Ideally, this approach is to be used in social networks where a set of Secondary Requesters have sparse meeting with Agents and Ad-Dealers but regular meeting with a user who can server as the Prime Requester. For example, Alice enters the proximity of Ad-Dealers regularly and also has regular meetings with Bob. Bob has regular meetings with Charlie and Dana who themselves hardly ever appear within proximity Ad-Dealers or any other users except Bob. Alice can therefore serve as Agent for Bob who can operate as a Prime Requester for Charlie and Dana.
5.2 Experiments

To test our Fragmented Profile Comparison (F-PC) framework, we implemented a series of simulations using Python. The first set of simulations, which is presented in Section 5.2.1, offers an estimate of the average rate of a successful selection of shared interests between two users. The second simulation in Section 5.2.2 measures the efficiency of the model in terms of total number of required attempts until a shared interest between two users is selected successfully. Our final simulation in Section 5.2.3 measures the expected bandwidth conservation of real data.

5.2.1 Shared Interest Selection Rate

In the experiments that are featured in the following sections, we simulate a series of F-PC profile comparisons between a requesting user \( R \) and an Agent \( A \). \( R \) and \( A \) have two non-identical Advertising Interest Profiles \( AIP_R \) and \( AIP_A \) respectively. The profiles are represented as a list of 400 interests which are marked as ‘TRUE’ or ‘FALSE’ depending on the user’s individual preferences. The goal is for \( R \) to select a 'TRUE' interest \( I \) out of \( AIP_R \) so that there is a high probability that \( I \) is also marked as 'TRUE' in the profile \( AIP_A \) of the Agent \( A \). We measure and compare the probability of \( R \) selecting a shared interest with \( A \) based on two methods. The first method represents our benchmark where \( R \) performs a simple Random Selection which mirrors the selection that would normally take place by a Requester who has no information about the Agent’s \( AIP_A \). The second method follows our F-PC (Fragmented Profile Comparison) framework where the Agent’s profile \( AIP_A \) is separated into fragments.

Experiment 1: Profiles Generated Randomly For our first experiment we used profiles which were generated randomly. In order to simulate a set of users with various types of consuming habits, we assigned to each user a total number of interests which ranges between 40 to 120 (10% to 30% of the total profile). As shown in Figure 5.3, for the Random Selection the success rate is on average of 20%. F-PC shows a steady increase as the number of fragments increases. For a profile of 4 fragments of 100 interests we have a success rate of 24.7%, for 20 fragments of 20 interests we have a rate of 37.7% and a peak rate of 48.9% for 40 fragments of 10 interests. These results are to be expected considering the fact that the profiles have been generated randomly and therefore have a uniform spread of marked interests.
F-PC (Fragmented Profile Comparison) with randomly generated profiles

**Experiment 2: Profiles Generated from Dataset** For our second experiment we utilize a dataset of Foursquare check-ins in 400 venues [154]. The venues represent the different advertising interests of the participating users based on their location habits. For example, a user who checked-in an airport is perceived as a consumer who is interested in traveling and a user who checked-in a university is perceived as a consumer who may be interested in student accommodations. As depicted in Figure 5.4, our results for the Random Selection show an average success rate of 31% which is not affected by the fragmentation of the profile. For the F-PC, we see
a steady increase when the profile is separated into smaller fragments. In more detail, we have a success rate of 37.6% for a profile with 4 fragments of 100 interests, 44.9% success for 20 fragments of 20 interests and a pic rate of 52.7% for 40 fragments of 10 interests. Comparatively to the results of the first experiment, we witness an increase in success rate. This is due to the fact that profiles which are generated based on real user habits show less uniformity in that way that marked interests are spread.

\( F\text{-}PC \) (Fragmented Profile Comparison) with profiled generated from a dataset

![Success rate contrast of \( F\text{-}PC \) (Fragmented Profile Comparison) and random selection with profiles generated from a dataset.](image)

Figure 5.4: Success rate contrast of \( F\text{-}PC \) (Fragmented Profile Comparison) and random selection with profiles generated from a dataset.
CHAPTER 5. PRIVATE PROFILE COMPARISON

Experiment 3: Profiles Generated Based on Pareto Principle  For our third experiment, we feature profiles which are composed based on the Pareto principle. The Pareto principle states that 80% of the output of a system is caused by 20% of the input. The Pareto principle is commonly used in marketing, sales and decision making [30]. Considering the wide approval of the Pareto principle, we make the assumption that it is possible to construct an AIP where 80% of the most popular advertising interests of a specific social network can be concentrated within 20% of the available fragments.

\( F-PC \) (Fragmented Profile Comparison) of profiles generated with Pareto principle

![Graph showing successful selection average per 100 comparisons for F-PC and random selection with profiles generated from Pareto principle.]

**Figure 5.5:** Success rate contrast of \( F-PC \) (Fragmented Profile Comparison) and random selection with profiles generated from Pareto principle.
CHAPTER 5. PRIVATE PROFILE COMPARISON

As depicted in Figure 5.5, the output of the Random Selection shows a success rate which starts at 31% but gradually decreases to 20%. This can be explained by the fact that the Pareto principle cannot be effectively applied to profiles that are separated in a small number of fragments. For example, when we have 2 fragments of 200 interests, the simulation will resolve into cramming into one fragments between 32 and 96 interests (80% of 40 and 120 which corresponds to the range of a user’s total interests). Depending on the way that the pseudo-random numbers are generated, the simulation may primarily focus on the one half of the profile that contains most of the user’s interests which would result in a high success rate.

The F-PC method yields a starting success rate of 36.6% for 2 fragments of 200 interests but rapidly increases to 64.3% for 4 fragments of 100 interests. This is to be expected as the top-ranked fragments contain between 32 and 96 interests which gives an average of 64. For the remaining selections, the simulation outputs a 78% success rate for 10 fragments of 40 interests and a pic rate for 93% for 40 fragments of size 10. These extreme results are not surprising since between 32 and 96 of the total interests are crammed in the top 8 or 9 fragments (20% of 40).

5.2.2 Delivery Efficiency

In this section we evaluate the delivery efficiency of F-PC in terms of the total number of times that an advert needs to be requested until it is eventually delivered. User $R$ performs a series of random encounters with set of Agents $A = \{A_1, A_2, \ldots, A_{100}\}$. Both the user and the Agents utilize a standard AIP of 400 interests which is separated into 20 fragments of 20 interests is each fragment. As in our previous experiments, the profiles of the Agents contain between 40 and 120 interests (10% to 30% of the total profile) which are marked ‘TRUE’. $R$ however starts with a profile AIP$_R$ which has all 400 interests marked as ‘TRUE’. During each encounter with an Agent, $R$ is allowed to make a single selection. The selected interest is marked as delivered (‘FALSE’) within AIP$_R$ only when it is shared by the encountered Agent. If that is not the case, the interest remains as ‘TRUE’ until it is re-selected with success. The intent of the experiment is to measure the number of selection attempts until every interest in AIP$_R$ has been selected successfully.

As show in Figure 5.6, F-PC yields 161 successful selections in the first attempt in contrast to Random Selection which only achieves 75. After the second attempt, F-PC has successfully selected 241 interests. This is matched by Random Selection with 250 selections only after the fourth at-
tempt at which point \textit{F-PC} has successfully selected 305 interests which corresponds to $3/4$ of the entirety of $AIP_R$. For the remaining attempts, \textit{F-PC} maintains a lead which is gradually reduced until the tenth attempt which results in \textit{F-PC} having selected 374 interests compared to 350 for the Random Selection.

These results indicate that a user who takes advantage of \textit{F-PC} will require fewer requests in order to receive his/her desired adverts. It has to be noted however that the present simulation is not entirely representative of the actual operation of the model as the \textit{Agents} are limited to serving \textbf{only} interests that they share with $R$. In an actual implementation of the model, this would not be the case as \textit{Agents} would also serve requested adverts for non-shared interests. Therefore, the results of the simulation are not indicative of the number of failed request attempts that will be endured by a user before his/her adverts are delivered but rather of the memory and bandwidth that will be conserved on the Agent’s device.

Delivery efficiency of \textit{F-PC} (Fragmented Profile Comparison)

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{delivery_efficiency.png}
\caption{Delivery efficiency of \textit{F-PC} (Fragmented Profile Comparison) in comparison to a random selection.}
\end{figure}
5.2.3 Resource Conservation

The experiments which were presented in the previous sections illustrate the conservation of memory and bandwidth by calculating the number of shared adverts. To assess how this conservation of resources translates to actual memory size, we performed a simple simulation of the encryption process with the use of real adverts. The adverts that we used were static PNG images of standard dimensions (300x250 pixels) and average size of 12 KB.

As \textit{ADS} was designed to support symmetric encryption, we used the AES (Advanced Encryption Standard) algorithm with a key size of 256 bits. As anticipated, the encrypted adverts measure the same size as the original images (12 KB) while the encrypted copy of the key only measured 4 KB, thus constituting to a 66.6\% conservation of memory for every advert of shared interest. In terms of run time, the encryption of each advert image required 0.464 seconds while the encryption of a key copy only required 0.182 seconds.

To also account for the prospective of adopting asymmetric encryption, we repeated the simulation with the use of the RSA (Rivest Shamir Adleman) cryptosystem. Results showed a significant increase in the size of the encrypted advert images to 16 KB while the key files also increased but only slightly, keeping them within the 4 KB threshold. Considering the fact the encrypted adverts require more memory that the originals, the conservation of memory is increased to 75\% for every advert of shared interest. Regarding run time, the encryption of each advert image required 0.717 seconds while the encryption of a key copy only required 0.448 seconds.

5.3 Evaluation

In accordance to the result of our experimentation and the findings of our qualitative analysis, we dedicate the following sections to conduct a privacy and performance evaluation of all three of the offered profile comparison algorithms.

5.3.1 Demographic Profile Comparison (\textit{D-PC})

\textit{Demographic Profile Comparison} (\textit{D-PC}) identifies advertising interests that two users may potentially share based on the similarity of their demographic attributes. Physical demographic attributes such as gender and age are openly shared and therefore simple to take into account. Social attributes, such as location and status, are however determined automatically based on
the social interactions of the two users. The effectiveness of \textit{D-PC} is therefore dependant on the availability of social demographic data, namely the meeting patterns and meeting locations of the two users. Users who interact for extended periods at locations of advertising value (locations which can be associated to specific advertising preferences) are expected to produce better results than users who meet more sporadically at locations of lesser advertising significance. Arguably, this limits the effectiveness of \textit{D-PC} but at the same time offers the highest level of privacy as no personal data is exchanged.

\subsection{Fragmented Profile Comparison (\textit{F-PC})}

\textit{Fragmented Profile Comparison (F-PC)} is designed to narrow the Requester’s selection on the fragments of the \textit{Agent’s AIP} which offer the highest concentration of marked interests. In contrast, a \textit{Random Selection} has a more broad field of focus as it targets the \textit{Agent’s AIP} in its entirety. Based on this premise alone, it is not surprising that \textit{F-PC} yields better results than a \textit{Ransom Selection} as the selection is performed out a smaller and more densely populated sample. One thing that is evident from our experiments however, is the fact that the effectiveness of \textit{F-PC} is relevant to the manner in which a \textit{AIP} is populated. On a randomly constructed \textit{AIP}, the effectiveness of \textit{F-PC} is significantly lower compared to \textit{AIPs} which were constructed based on a real user dataset or the Pareto principle. The deviation in effectiveness can be explained when we consider the way that marked interests are distributed within the \textit{AIP} in each case. A randomly generated \textit{AIP} produces a more uniform distribution which results in all fragments having a similar number of marked interests. Contrary to that, and \textit{AIP} of non-uniform distribution produce fragments of either very high or very low numbers of marked interests. From a practical point of view, a non-uniform distribution could stem when neighbouring entries within the \textit{AIP} represent related advertising interests. For example, a user who is very active in sports would be likely to rank very highly a fragment that contains the entries for sport shoes, truck suits and activity monitors. An \textit{AIP} of such design offers better effectiveness but also presents a potential privacy threat as an \textit{Agent’s} top ranked fragments could reveal interest in a particular consumer field. The structure of the \textit{AIP} is therefore left in the discretion of the system administrator based on the desired balance between effectiveness and privacy.
5.3.3 Selective Profile Comparison (S-PC)

Selective Profile Comparison (S-PC) follows a cryptographic approach in order to allow an Agent to select the best option between a series of Candidate Selections CSs. The Agent can see the exact number of shared interest within each CS which practically means that the effectiveness of S-PC is dependant on the total number of performed CSs. In regards to privacy, it can be argued that S-PC is less secure than alternative profile comparison methods as the Agent learns the number of shared interests in each CS. For a malicious Agent it would even be possible to determine if the Requester has a particular interest by fabricating his/her profile so that only one interest is marked. Consequently, the malicious Agent would be certain that the victimized Requester has the particular interest marked if any of the weights of the CSs is equal to ‘one’. What renders the aforementioned attack even more dangerous is the fact it can be performed stealthily. The Requester is also able to perform a similar attack by fabricating two CSs which differ only by a single interest. As a valid example, consider a scenario where a malicious Requester $\hat{R}$ composes $CS_1=[I_1 \ I_2 \ I_3]$ and $CS_2=[I_1 \ I_2 \ I_3 \ I_4]$. Note that $CS_2$ differs to $CS_1$ only by a single interest which is $I_4$. Consequently, if the particular interest is marked by the victimized Agent, the weight of the corresponding $CS_2$ will be higher and would therefore be chosen. Collaborative-Selective Profile Comparison (CS-PC) can potentially mitigate the limitations of S-PC but at the same time introduces certain practical shortcomings. Firstly, CS-PC requires multiple meetings and cannot therefore be performed in real time. Secondly, the Prime Requester $R_0$ needs to temporarily consume additional memory to accommodate for the data of Secondary Requesters.

5.3.4 Overall Evaluation

An evaluation summary of the various profile comparison methods is shown in Table 5.1. The designations 'Good’ and 'Poor’ respectively indicate that an approach offers optimal performance or entirely fails at a particular field. The designation 'Adequate' indicates that the performance of an approach is not optimal but still substantial while the designation 'Limited' indicates that the approach offers some level of performance which is however insufficient. Lastly, the designation 'Tunable' is used to describe an approach which can exhibit different levels of performance (ranging from 'Poor’ to 'Good’) depending on the selected input parameters. Demographic Profile Comparison (D-PC) is easy to implement and it does not require the ex-
change of any personal information. \textit{D-PC} therefore offers a 'Good' level of practicality and privacy for both participating users but also has a Limited effectiveness which is conditional on the availability of social demographic data. \textit{D-PC} is ideal for users who have regular social interactions but are reluctant to share any information about their consumer preferences. \textit{Fragmented Profile Comparison (F-PC)} grants a 'Tunable' compromise between privacy and effectiveness for Agent and 'Adequate' privacy for the Requester who does not need to share any information about his/her profile. \textit{F-PC} also offers great flexibility for the Agent who is free to configure the system in order to achieve a desired compromise between effectiveness and privacy. \textit{F-PC} is the most preferable option when the priority is Requester privacy while the Agent is willing to make a compromise between effectiveness and privacy. \textit{Selective Profile Comparison (S-PC)} offers a 'Tunable' effectiveness level which is contingent to the number of \textit{Candidate Selections CSs}. Given enough CSs, \textit{S-PC} can achieve optimal effectiveness but this will come as a trade off to the privacy of the Requester. Furthermore, \textit{S-PC} shows 'Limited privacy as it' is susceptible to attack from either one of the participants. \textit{S-PC} is therefore recommended for interactions where there is a partial level of trust between the two users. \textit{Collaborative-Selective Profile Comparison (CS-PC)} introduces additional Requesters to \textit{S-PC} which increases privacy to 'Tunable' for all participants but it also adds complexity which makes it 'Poor' in terms of practicality. \textit{CS-PC} is recommended for large communities of users who have limited access to Agents and Ad-Dealers.

<table>
<thead>
<tr>
<th></th>
<th>Privacy</th>
<th>Effectiveness</th>
<th>Practicality</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-PC</td>
<td>Good</td>
<td>Limited</td>
<td>Good</td>
</tr>
<tr>
<td>F-PC</td>
<td>Tunable</td>
<td>Tunable</td>
<td>Adequate</td>
</tr>
<tr>
<td>S-PC</td>
<td>Limited</td>
<td>Tunable</td>
<td>Adequate</td>
</tr>
<tr>
<td>CS-PC</td>
<td>Tunable</td>
<td>Tunable</td>
<td>Poor</td>
</tr>
</tbody>
</table>

Table 5.1: Evaluation table of profile comparison methods.

5.4 Private Profile Comparison Summary

In this chapter we call attention to the fact that opportunistic networks typically suffer from an wasteful utilization of resource. To resolve this issue for
the opportunistic network of ADS we offer an scheme which allows multiple users to cooperatively request and access the same encrypted adverts. To enable users to identify adverts of shared interest within their respective advertising profiles, we propose a series of profile comparison algorithms: D-PC, F-PC, S-PC and CS-PC. The key innovation of all four designs lies on the fact that they are built from the ground up to maintain user privacy. To evaluate our algorithms, in both terms of accuracy as well as privacy, we performed a series of qualitative evaluations and experimental simulations. The results demonstrate that each algorithm is capable of achieving a varied balance between accuracy and privacy which offers a great deal of flexibility to the users of ADS who can either prioritize on resource utilization or privacy in accordance to their individual needs.
Chapter 6

ADS+R: Advert Fraud Prevention

ADS+R (Advert Distribution System with Reporting) was published in [98] as an extension of ADS. ADS+R utilizes the same infrastructure as ADS but also introduces the concept of Behavioural Verification as a novel approach for preventing advert fraud while still maintaining user privacy. ADS+R accomplishes both fraud prevention and user privacy by incorporating client-side processing and a blockchain-inspired architecture which enables mobile users to compose verifiable Ad-Reports and submit them without exposing their identities. Although the majority of users have no immediate benefit from submitting fraudulent Ad-Reports, this does not ensure that every filled report corresponds to real consumer activity. As users need to remain anonymous, identifying dishonest (malicious) reports through traditional methods such as digital signatures is undesired. To address this limitation, our contribution is a mechanism which enables the verification of reports that were submitted by honest users without compromising the user’s identity. What constitutes users as honest or dishonest is the manner by which they access adverts on their mobile devices. Dishonest users commit fraud by submitting multiple fake reports over a short period of time while honest users operate under the scope of consumers who view adverts at a balanced pace while engaging in typical social activities such as making online purchases, moving through space and interacting with other mobile users.

We argue that it is hard for dishonest users such as clickbots and click-farms to fake honest behaviour and we exploit the behavioural patterns of users in order to classify Ad-Reports as real or fabricated. ADS+R composes an anonymous log of the user’s behavioural patterns which allows Advertisers
to determine her honesty by detecting anomalies such as deficiency of advert
engagement, lack of mobility and social interaction and unrealistically large
volumes of traffic over short periods of time. In contrast to previously pro-
posed systems, \( ADS+R \) offers a more secure reward-claiming model which
protects against fraud while still preserving user anonymity. To the best of
our knowledge, our system is the first to (1) successfully exploit behavioural
patters for the purpose of exposing advert fraud while (2) still preserving
user privacy and contrast to alternative methods, our approach (3) does not
require complex filtering to identify reports which originate from the same
source.

In Section 6.1 we offer the system specifications of \( ADS+R \) and then in
Section 6.2 we provide a detailed overview of the system. In Section 6.3 we
summarize the operation of the protocol and we finally evaluate our design
in Section 6.4.

6.1 System Specifications

In the following sections we offer the system specification of \( ADS+R \). In
Section 6.1.1 we provide a high level description of the system’s architecture
and in Section 6.1.2 we describe our trust model. Finally in Section 6.1.3
we determine our system requirements which will also serve as evaluation
criteria for our design.

6.1.1 System Architecture

\( ADS+R \) is an extension of \( ADS \) that was presented in Chapter 4 and therefore shares much of the same components and architecture. Users, Publishers, Advertisers, Ad-Dealers and Broker represent the same stakeholders as in \( ADS \) but at the same time have the added functionality of managing Ad-Reports. Users generate Ad-Reports which are forwarded to the Broker via Ad-Dealers. The Broker is responsible for validating Ad-Reports on behalf of Advertisers via a mechanism which we term Behavioural Verification. Once reports have been validated, Advertisers reward the Ad-Dealers and Publishers for their services. Users can directly interact with Ad-Dealer or establish opportunistic connections via Agents who can ferry Ad-Reports in addition to adverts, ARMs (Advert Request Messages) and DMs (Delivery Messages). Both functions can be performed at the same time and by the same Agent. Consider a simple example where Alice uses an Agent Bob to transfers both ARMs and Ad-Reports. At the same time, Bob can be transferring ARMs for Charlie and additional Ad-Reports on behalf of Danna.
Upon contacting an Ad-Dealer, Bob can perform both actions of submitting the Ad-Reports and collecting adverts during a single session.

Figure 6.1 illustrates the complete architecture of ADS+R and provides a high-level overview of the system’s operation which can be divided into two stages (sub-protocols) which are detailed in the following paragraphs.

**RF (Report Form) Issuing:** The user initiates the first stage by composing an RF-Request, the purpose of which is to inform the Broker of his/her intention to submit Ad-Reports. The RF-Request is encrypted with a public key which belongs to the Broker and sent to an Agent who physically transfers the RF-Request to one of the Ad-Dealers. The Ad-Dealer forwards the
RF-Request to a Broker who decrypts it and issues an RF (Request Form). The RF is encrypted with a key which was provided to the Broker within the RF-Request and the RF is then sent back to the Ad-Dealer. Lastly, the Ad-Dealer forwards the RF to the Agent so that it may be conveyed back to the requesting user. The RF contains information which is needed by the user in order to compose and submit Ad-Reports.

Ad-Report Submission: The second stage takes place after the user has run ADS and has viewed his/her adverts through a Publisher. To notify the Broker that the adverts have been viewed, the user composes Ad-Reports and forwards them to an Agent (not necessarily the same Agent that was used in the first stage) with the intent to be delivered to an Ad-Dealer. The Ad-Reports are encrypted with the Broker’s public key and therefore not accessible to the Agent or Ad-Dealers. The receiving Ad-Dealer shares the Ad-Reports with the Broker, who verifies their authenticity and notifies the appropriate Advertisers. Based on the information that is provided within the Ad-Reports, the Advertisers can reward the Publishers who featured the adverts as well as the two involved Ad-Dealers (the Ad-Dealer who submitted the Ad-Reports but also the one that was responsible for delivering the adverts to the user via ADS).

6.1.2 Trust Model

The trust model we assume for ADS+R is an upgraded version of the trust model which we used for ADS in Section 4.1.2. The Broker is therefore still regraded as trusted by the Advertisers and the trust relations which concern the users remain the same. Users are still regarded as non-malicious since they have no immediate benefit to commit fraud. However, it is possible for an adversary to submit fictitious Ad-Reports by assuming the identity of a user without being exposed. The Broker and Advertisers are therefore not threatened by the users but from Ad-Dealers and Publishers who may assume the identity of a user in order to commit fraud. For that reason, Ad-Dealers and Publishers but not users are upgraded to malicious in the eyes of the Broker and Advertisers. The Advertisers are therefore willing to accept the authenticity of an Ad-Report only after it has been verified by the Broker. The Broker and Advertisers on their part can benefit from altering the content of an Ad-Report in order to avoid paying a commission. Ad-Dealers and Publishers have therefore no reason to trust the Broker and Advertisers and also consider them as malicious. The updated trust relations between the system’s stakeholders can be seen in Table 6.1 with
the changes marked in bold letters.

<table>
<thead>
<tr>
<th></th>
<th>Advertisers</th>
<th>Broker</th>
<th>Ad-Dealers</th>
<th>Users</th>
<th>Publishers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advertisers</td>
<td>-</td>
<td></td>
<td>Malicious</td>
<td>-</td>
<td>Malicious</td>
</tr>
<tr>
<td>Broker</td>
<td>-</td>
<td>Trusted</td>
<td>Malicious</td>
<td>Malicious</td>
<td>Malicious</td>
</tr>
<tr>
<td>Ad-Dealers</td>
<td>Malicious</td>
<td>Malicious</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Users</td>
<td>Curious</td>
<td>Curious</td>
<td>Curious</td>
<td>Malicious</td>
<td>-</td>
</tr>
<tr>
<td>Publishers</td>
<td>Malicious</td>
<td>Malicious</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6.1: Table of trust relations between ADS+R stakeholders.

6.1.3 System Requirements

Having considered the trust relations between stakeholders in Section 6.1.2, we dedicate this section to compose an index of system requirements that will serve as the criteria under which the effectiveness and security of our design can be evaluated.

- **Reporting Effectiveness:** Ad-Reports should include all necessary information to ensure that participating stakeholders are able to effectively claim their rewards. The Broker should be able to ensure that each report is accounted only once and also there should exist a way for users to confirm that a report was delivered successfully.\(^1\)

- **Reporting Fraud Prevention:** The Broker should be able to prevent any group of conspiring Ad-Dealers and Publishers from submitting reports which do not correspond to real consumer activity.

- **Reporting Integrity:** It should not be possible for the Broker or Ad-Dealers to alter the content of a submitted Ad-Report for the purpose of deceiving each other.

- **User Privacy:** A user’s sensitive information should remain private from all parties, including other users.

\(^1\)The effectiveness of submitted reports is taken as a standard requirement by analogous systems but in the case of our model it needs to be examined in more detail as it may be affected by the additional mechanisms that are used to preserve privacy (opportunistic networks and anonymous submission).
6.2 System Overview

In the following sections we provide an analysis of ADS+R and offer a detailed insight into our method for detecting fake Ad-Reports without any need for knowing the identity of the submitting users. Our design is based on a novel approach which we have termed as Behavioural Verification. We argue that honest user behaviour is hard to fake by dishonest users such as bonets and click-farms. Behavioural Verification exploits typical social behavioural patterns in order to verify honest users without knowing their identities.

As users view adverts on their devices, they generate Ad-Reports as featured in Section 6.2.2. At the same time, users also collect a series of Tokens when they perform certain daily activities such as purchasing goods, visiting different locations or interacting with other users, as explained in Section 6.2.5. To better comprehend the conceptual idea of Tokens, think of a game of scavenger-hunt where players can prove to have performed a required task (e.g., solved a puzzle or visited a location) by recovering some type of artifact.

Tokens are then linked to Ad-Reports in the form of a blockchain-inspired construction which is termed as ARC (Ad-Report Chain) and is further analyzed in Section 6.2.3. As the ARC contains both the user’s Ad-Reports and Tokens, it can be used by the Broker to verify that the Ad-Reports were submitted by a user who exhibits honest social behaviour. Furthermore, Tokens work as time-stamps which allow the Broker to verify that the Ad-Reports of an ARC were created at a paced rate and not in bulk (as the creation of bulk amounts of Ad-Reports in short time is indicative of fraudulent behavior). The Broker is responsible for validating the honesty of submitted ARCs and notifies the Advertisers so that commissions can be awarded to the concerned Ad-Dealers and Publishers. To maintain user privacy, ARCs are encrypted and only visible to the Broker (and by association to Advertisers) however, Ad-Dealers and Publishers are able to audit the Broker’s integrity with the use of cryptographic hash functions. Information among the system stakeholders is shared through the use of a digital database termed as the SC-Board (Service Confirmation Board) as specified in Section 6.2.4.

6.2.1 System Setup

For the submission of Ad-Reports, ADS+R utilizes a different set of cryptographic keys than those used for the delivery of adverts. Users are required
to encrypt their ARCs with a public key $BroK^{Pub}$ which is also pre-installed on their clients while the corresponding private key $BroK^{Pri}$ is only known to the Broker. The Broker also issues a different Token Singing Key $ToK^{sig}_{Aid}$ (e.g., $ToK^{sig}_{1}$, $ToK^{sig}_{2}$, ..., $ToK^{sig}_{n}$) to each individual Ad-Dealer. Ad-Dealers keep their $ToK^{sig}_{Aid}$ private from each other as to ensure security. Token Singing Keys are used for the signing of Tokens which can later be verified by the Broker who has access to the corresponding Token Verification Keys $ToK^{ver}_{Aid}$.

### 6.2.2 Ad-Reports

$ADS+R$ offers three different types of Ad-Reports which can support all of the available pricing models that are used in traditional OBA. The different pricing models were described in detail in Section 2.1 but are also briefly described below along their matching Ad-Report types.

- **RoV (Report of View):** RoV is used to support the PPM Pay-Per-Mille model which grants an award when an advert is viewed by a user.

- **RoC (Report of Click):** RoC is used to support the PPC Pay-Per-Click model which grants an award when an advert is clicked by a user.

- **RoA (Report of Action):** RoA is used to support the PPA Pay-Per-Action model which grants an award when a specific action is performed by a user after an advert has been clicked.

As depicted in Figure 6.2, all supported Ad-Report types incorporate a sequence number $N$ which indicates the order in which the reports were created. The sequence number is what allows the system to link reports into a blockchain-inspired architecture and it is therefore imperative that each generated report contains the correct $N$. The Advert Code is a unique reference number that is sent to the user alongside each advert. The $A_{id}$ has been analyzed before in Section 4.2.1 and accommodates the identity of the Ad-Dealer who distributed the advert. Respectively, $P_{id}$ represents the identity of the Publisher who featured the advert to the user while the Date field holds the date and time of the publication.

The C-Token (or Click-Token), which can be found in the RoC and RoA, is a sequence of data which can be obtained by the user when an advert has been clicked. The A-Token (or Action-Token) which is present in the RoA,
follows a very similar format as the C-Token with the main difference being that it is disclosed to a user only after a specific condition has been met (e.g., the user made a purchase). Each Advertiser periodically generates their own C-Token and A-Token which are uploaded within their domain. The function which is used for this operation as well as the frequency upon which the two tokens are updated fall under the responsibility of the respective Advertiser. Ideally, the C-Token and A-Token should be generated by a cryptographically secure random number generator and as often as practically possible\(^2\). A design feature which is similar to Tokens is also presented by Juels et al. [81] where the authors make use of cryptographic credentials known as Coupons.

The rate at which the C-Token and A-Token are updated influences the system’s accuracy of verifying the time that an Ad-Report was created. More specifically, if Tokens are updated once every \(T\) time units, then the ADS+R can verify the time of a user’s report with granularity \(T\). The C-Token is uploaded in the same cyberspace where the user is linked to when clicking on the advert while the A-Token is placed in the location to which the user is diverted to when they perform a specific action such as making a purchase. Much like the way that web cookies work, the mobile client obtains the C-Token and A-Token from the Advertiser’s website and places them within the Ad-Report as the user is browsing. This enables Advertisers to verify that a user accessed their website or performed a specific action before creating a RoC or RoA. Having to obtain Tokens before creating a new Ad-Report, makes the forging of RoCs and RoA more difficult. To forge a RoC, the dishonest user needs to fist visit the Advertiser’s web site while forging a RoA requires the performing of an action. More importantly, Tokens prevent dishonest users from creating fictitious Ad-Reports ahead of time as a RoC or RoA can be created only after the contained C-Token and A-Token have been made available. Lastly, \(h(AR_{(N-1)})\) contains a hash function digest of each previous Ad-Report that was composed by the same user. This enables users to link all of their Ad-Reports in the form of a blockchain-inspired architecture which is analyzed more minutely in Section 6.2.3.

One last thing that needs to be mentioned is the fact that in every Ad-Report, the sequence number \(N\) and hash \(h(AR_{(N-1)})\) are sent in plaintext form while the remaining fields are encrypted with the Broker’s public key.
Figure 6.2: Supported types of Ad-Reports and their contented elements.

Rather than dealing with individual Ad-Reports as they are being created, ADS+R enables users to aggregate multiple Ad-Reports throughout the course of a defined period and then submit all of them as a single unit. As it has already been explained in Section 6.2.2, each Ad-Report contains the hash digest \( h(AR_{(N-1)}) \) of the previous Ad-Report \( N - 1 \). This enables the user to link several Ad-Reports together in a form that resembles the architecture of a blockchain and is termed as the ARC (Ad-Report Chain).

As shown in Figure 6.3, the first block of the ARC contains an initiating value which is marked as ARC-ID. The ARC-ID is hashed to produce \( h(ARC - ID) \) that is included in the second block \( N = 1 \) with each consecutive block following the same arrangement. The \( h(ARC - ID) \) essentially works as a unique identifier which also marks the start of a specific ARC. The ARC-ID is dictated by the Broker and sent to the user within the RF (Report Form) as depicted in the same figure. Recall from Section 6.1.1 that the RF (Report Form) is a message that comes as a response to the user’s request to file Ad-Reports.
CHAPTER 6. ADS+R: ADVERT FRAUD PREVENTION

In addition to the ARC-ID, the RF also contains a cryptographic Report Signing Key $RepK_{user}^{sig}$. While the ARC-ID is used to identify and mark the start of an ARC, $RepK_{user}^{sig}$ is used to mark the end in such a way that the removal or addition of blocks to a submitted ARC is prevented. More specifically, the user calculates the hash digest $h(ARC)$ of the ARC and then signs it with $RepK_{user}^{sig}$ in order to produce an Integrity Hash IH which can also be seen in the same in Figure 6.3. To confirm that the IH was created by the user, the Broker can verify it with the use of the secret verification key $RepK_{user}^{ver}$ and she can then compare the $h(ARC)$ from within the IH to an $h'(ARC)$ which the Broker computes herself in order to verify that the ARC has not been altered.

Figure 6.3: Structural information components of ADS+R.
6.2.4 SC-Board (Service Confirmation Board)

The *SC-Board* (Service Confirmation Board) is a digital database which serves as an information sharing platform between all stakeholders of *ADS+R*. The indexed entries of the *SC-Board* represent *RFs* (Request Forms) that have been distributed to users and consist of five fields as shown in Figure 6.4.

The first two fields are input by the *Broker* when she issues a new *RF* and respectively contains the *ARC-ID* and the identity $A_{id}$ of the issuing Ad-Dealer (the Ad-Dealer who forwarded the user’s *RF-Request*) along with the corresponding date. The remaining fields are completed when the *ARC* is submitted with the third field keeping the identity $A_{id}$ of the submitting Ad-Dealer (the Ad-Dealer who forwarded the user’s *ARC* and *IH*) and the date of submission while the fourth and fifth contain the Ad-Report Chain *ARC* and Integrity Hash *IH*.

Indicated in the diagram with a darker shade under the second, third and fourth field, are certain sections which are completed by the issuing *Ad-Dealer*, the submitting *Ad-Dealer*, the *Broker* and the individual *Advertisers*. These fields serve the purpose of verification checks. In more detail, *VC-I* under the second field is signed by the issuing *Ad-Dealer* to verify the issue of the new *ARC-ID*. In a very similar fashion, the submitting *Ad-Dealer* signs the third field marked as *VC-S* in order to verify the submission of the *ARC* and confirm the correctness of the hash digests $h(N)$ of all blocks (*Ad-Reports*). Recall that in Section 6.2.2 we briefly mentioned that the content of *Ad-Reports* is encrypted except for the sequence number $N$ and hash $h(N)$ which are still visible to the submitting *Ad-Dealer*. While the *ARC* in the fourth section is published by the *Broker* after decryption, the submitting *Ad-Dealer* confirms that the hashes have not been altered by comparing them to his own copy. The individual verification checks, which are marked as *VC-1* to *VC-N* under the *ARC*, are filled either by the *Broker* to indicate blocks that have been verified or by the *Advertisers* to indicate blocks for which an *Advertiser* has awarded a commission to the respective *Publisher*. More details on the exact operation and the reasons behind these verification checks are provided in Sections 6.3 and 6.4. Lastly, we need to mention that all fields of the *SC-Board* are visible to *Ad-Dealers*, *Publishers* and *Advertisers* but the first field which shows the *ARC-ID* also becomes available to users after submission has been completed. Users only need to have access to the first field in order to verify that their submission has been delivered but cannot see any other information that is published on the *SC-Board*. 
6.2.5 Behavioural Verification

The detection of forged Ad-Reports is a challenging issue because users need to remain anonymous, and anonymity prevents verification through traditional methods such as digital signatures. To resolve this problem, we propose an alternative means of verifying truthful reports while still allowing users to maintain their anonymity. Users can be classified as honest or dishonest based on the manner upon which they create Ad-Reports. As Ad-Reports are rewarded at a low commission (typically at around $1 per 1000 impressions), dishonest users commit fraud by generating large volumes of unverifiable Ad-Reports at a rate which is much higher than what is realistically possible for a legitimate consumer. Honest users on the other hand, view adverts at a realistic rate and therefore generate Ad-Reports in a paced manner over a longer time period. While composing their Ad-Reports, honest users engage in typical social activities such as purchasing goods, moving through space and interacting with other users. All of these social activities are distinguishing behaviours of honest users which can be exploited to

\[ h(ARC) \]

\[ \begin{array}{c|c|c|c|c|c|c} 
\text{ARC-ID: 1} & \text{Issuing Ad-Dealer} & \text{Submitting Ad-Dealer} & 1 & 2 & 3 & N \\
\text{Date} & \text{Date} & \text{Report} & h(N) & VC-1 & & \\
\text{VC-1} & \text{VC-5} & & & & & \\
\end{array} \]

\[ \text{ARC-ID: #} \]

Figure 6.4: Visual representation of the CS-Board (Service Confirmation Board).

\[ ^3 \text{Fraud that is committed at a limited scale (for e.g. users who periodically click adverts with non-consumer intent) is practically unfeasible to detect but also yields trivial returns.} \]
verify legitimate Ad-Reports.

As we already described in Section 6.2.3, Ad-Reports that are created by the same user are linked together in an ARC. The goal is therefore to identify whether the creator of a particular ARC is honest or dishonest. We accomplish this by embedding into the ARC certain elements (blocks) which reveal the user’s social behaviour patterns during the time that Ad-Reports were being created.

**Advert Association**

Honest users utilize adverts as consumers and are therefore likely to not simply view an advert but to also engage with it by clicking or making a purchase. The act of engaging with an advert can therefore be considered as a typical behaviour of honest users but it also has to be noted that not all honest users engage with adverts in the same rate, and some users do not engage at all. In order to therefore avoid false positives, ADS+R regards the engagement with adverts as an indicator of honesty but the lack of engagement is not treated as suspicion of dishonesty. To compensate for users who do not engage adverts, ADS+R exploits other forms of honest behaviour as explained in the following sections.

In Section 6.2.2, we illustrated the available types of Ad-Reports and called attention to the fact that a RoA is harder to forge than an RoC which is in turn harder to forge than an RoV as the required C-Tokens and A-Tokens can be acquired only after accessing an Advertiser’s domain. The RoCs and RoAs can therefore serve as indicators of honesty as they signify that a user took the time to visit an Advertiser’s website. The remaining RoVs are not verifiable but can be validated by association when in the same ARC as shown in Figure 6.5.

A limitation to this approach lies in the fact that RoCs and RoAs are designed to be used by Advertisers who support the Pay-Per-Click (PPC) and Pay-Per-Action (PPA) advertising models. This may limit the number of RoCs and RoAs as it excludes all the Advertisers who only support Pay-Per-Impression (CPM). To overcome this shortcoming, ADS+R utilizes the different types of Ad-Reports (RoV, RoC and RoA) not based on the Advertiser’s pricing model but in accordance to a user’s engagement with an advert. Consider a simple example where an Advertiser supports CPM which means that a simple RoV would normally suffice. For the same application however, we can also use a RoC or a RoA when the user interacts with the advert by clicking or by performing an action. The commission is still going to be awarded based on the viewing but the use of a more secure
Figure 6.5: Example of Behavioural Verification through advert association.

*Ad-Report* type adds validity to the authenticity of the claim (as *RoCs* and *RoAs* are harder to forge than *RoVs*).

**Time and Location Checkpoint**

*Tokens* which are used in *RoCs* and *RoAs* are indicators of honest behaviour as they demonstrate that a user invested time into performing a specific action but they can also be used to determine the rate at which the *Ad-Reports* of an *ARC* were created (as *Tokens* are periodically updated). However, as not all users engage with adverts regularly enough for this method to be effective on its own, the same principle can be extended by periodically incorporating into the *ARC* some form of Time-Token (*T-Token*) which can signify the time that a particular block was created. One limitation that needs to be considered however, is that this *T-Token* cannot be obtained through the internet, as this would expose the user’s IP address and it would also be ineffective since a dishonest user could commit fraud by creating multiple *ARCs* in parallel over a longer period of time.

To overcome this limitation, the *T-Token* is distributed directly from *Ad-Dealers* in the same way as adverts. *T-Tokens* enable the *Broker* to verify the time that a block of the *ARC* was created but also operates as a location tag. Location tags are data that can be associated with a point in space and time and have appeared in the literature before, in the context of private (cryptographic) proximity testing in [108]. The location tag provides additional proof of a user’s honesty as it verifies the user’s social behaviour in terms of appearing within proximity of public locations, where *Ad-Dealers* are broadcasting. To further comprehend this notion, consider the following example of attempted forgery. If the *T-Token* were to be accessible online, a dishonest user \( \hat{U} \) could periodically download it and use it to easily verify set \( \hat{S} = \{ \hat{ARC}_1, \hat{ARC}_2, ..., \hat{ARC}_i \} \) of fictitious *ARCs* over a longer period of time. However, when the *T-Token* is distributed by *Ad-Dealers*, it is more difficult for \( \hat{U} \) to validate multiple *ARCs* since it requires them to physically travel to the location of an *Ad-Dealer* and request multiple *T-Tokens* for
each of the elements of \( \hat{S} \). Furthermore, so as not to raise suspicions, the T-Tokens would also need to be requested at a slow rate and preferably from different Ad-Dealers which adds a supplementary layer of difficulty for \( \hat{U} \).

When entering the vicinity of an Ad-Dealer, users can send a Token Request Message TRM. The TRM is encrypted with the System Encryption Key SysEK and contains the hash digest \( h(AR_{n-1}) \) of the last block of the user’s ARC and a user-generated symmetric encryption key \( K^{user} \). The Ad-Dealer decrypts the TRM with the System Decryption Key SysDK and begins to compose a Checkpoint Block (CB). As illustrated in Figure 6.6, the CB contains the Ad-Dealer’s identity \( Aid \), the user’s hash digest \( h(n - 1) \) and a time-stamp both signed with his Token Singing Key \( ToK^{sig} \). The time-stamp serves as the T-Token while the Ad-Dealer’s signature serves as proof of the user’s location. Before being sent back to the user, the CB is first encrypted with the Broker’s public key \( BroK^{Pub} \) and then encrypted again with the user’s encryption key \( K^{user} \) as shown in Equation (6.1):

\[
E_{K^{user}}[E_{BroK^{Pub}}[CB]] \tag{6.1}
\]

When the cryptogram is received, the user decrypts it with his/her own copy of \( K^{user} \) and obtains \( E_{BroK^{Pub}}[CB] \) which is given a sequence number \( N = n \) and is inserted into the ARC as shown in Figure 6.6. \(^4\)

\(^4\)All blocks of the ARC are encrypted with the Broker’s public key \( BroK^{Pub} \) as to ensure privacy against Ad-Dealers and Agents. The encryption on the CB could have been performed by the user but in this case is performed by the Ad-Dealer in order to relieve some of the strain from the user’s mobile device. Ad-Dealers can be trusted with this operation as they have no benefit from providing a defective CB.
Social Affiliation

The social affiliations between honest users is an additional behaviour which can be exploited to verify the rate that an ARC was created. When two users meet, they may exchange ARC-IDs as well as the sequence number $N$ and hash $h(n)$ of their last blocks. The two users can then verify the date and time of the meeting (T-Token) by adding an Affiliation Block $AB$ in their respective ARCs with each others’ information. In order to be valid, the $AB$s which are added to the ARCs of both users need to have matching T-Tokens but this does not require perfect synchronization. Mobile applications typically have a recommended refresh rate for adverts that is between 30 to 120 seconds while automated clickers and Click-Farms generate fake Ad-Reports at a much higher rate. For the purpose of detecting fraud, the time difference between the two users can therefore be tolerant to a margin of a couple of minutes without seriously affecting the system. In the event that two $AB$s do not match because one of the users provided an inaccurate date and time (either maliciously or accidentally), the Broker can simply ignore it while relying on other Tokens to validate the particular ARC.

Figure 6.7 illustrates an example where two users $A$ and $B$ have added each others’ Affiliation Blocks within their respective ARCs. The $AB$ which was added by user $B$, is shown in the diagram to contain a new sequence number $n$, the hash of the previous block $h(n - 1)$, the T-Token of the meeting (as registered by $B$) and the information that was sent by $A$ which includes her $ARC-ID=ARC-1$ as well as the sequence number $m$ and digest $h(m)$ of her last block. The date which is registered in $B$’s $ARC-2$ works as the T-Token which verifies the last block of $A$’s $ARC-1$ at a particular time. Notice that $ARC-1$ and sequence number $m$ are sent encrypted with the Broker’s public key $BroK^{Pub}$ while the $h(m)$ is signed with $A$’s Report Signing Key $RepK^{sig}_A$. This ensures that $B$ does not learn any information about $ARC-1$ nor is she able to alter $h(m)$.

Through the exchange of $AB$s, the Broker can infer that two ARCs were submitted by affiliated users but this does not compromise user privacy. ARCs are submitted anonymously and the Broker has no means of obtaining any information about a particular user’s social network nor is he able to identify ARCs that were submitted by the same user. However, one limitation of this verification method lies on the fact that a dishonest user may be able to verify multiple fictitious $ARC$s by exchanging $AB$s. Although plausible, this is prevented by combining all three verification methods as discussed in the following Section 6.2.5.
CHAPTER 6. ADS+R: ADVERT FRAUD PREVENTION

Figure 6.7: Example of Behavioural Verification through the use of AB (Affiliation Block).

Composite Verification

The individual methods of *Behavioural Verification* have certain limitations. The social affiliation approach in Section 6.2.5 is susceptible to fraud by means of creating multiple fictitious \( ARCs \) while the methods which are described in Sections 6.2.5 and 6.2.5 may not always be practically feasible as they require user to regularly click on adverts or travel to certain locations.

To compensate for each others’ limitations, all three approaches were designed to work in combination. In the example which is provided in Figure 6.8, user \( A \) submits an *ARC* which contains multiple *Ad-Reports* that need to be verified (marked in the figure with exclamation marks). The honesty of \( A \) is supported by the fact that his *ARC* also contains a verifiable report (either a *RoC* or a *RoA*), a *Checkpoint Block CB* from an *Ad-Dealer* and two *Affiliation Blocks ABs*.

Furthermore, we see that the respective *ARCs* of the two users \( X \) and \( Y \) who provided *ABs* for \( A \) also have verifiable reports, *CBs* as well as *ABs* from other users who also have their own verifiable credentials. As all submitted *ARCs* show indications of social activity, it serves as significant evidence to support the notion that they were composed by different honest users rather than a single dishonest one. For reasons of simplicity, the ex-
ample just described features only a few verification credentials. However, in a more realistic scenario, the users would likely have multiple credentials which would solidify their verification as the product of genuine social activity.

Figure 6.8: Example of Behavioural Verification through the combination of all available methods.

6.2.6 Statistical Analysis of Consumer Behavioural Patterns

In addition to combating advertising fraud, ADS+R can also prove to be usefully for purposes of market research. ARCs offer invaluable insight on the consumer preferences and social behaviour patterns of the submitting users while still preserving the user’s privacy. By inspecting the Tokens of
a submitted ARC, a market analyst would be able to infer complex correlations between advertising interests, location habits and social affiliations. For example, users who visit location X are likely to be interested in product A and users who are interested in product B are likely to affiliate with users that are interested in product C. Such insight will allow marketers to fine-tune their targeting algorithms and consequently increase the system’s effectiveness. To collect consumer information, marketers currently need to rely on analytic companies who gather their data through the use of tracking and questionnaire forms. Undoubtedly, such practices are threatening for user privacy and have questionable reliability. The introduction of ADS+R in market research will likely increase data quality, reduce the cost and complexity data acquisition and most importantly it will ensure user privacy.

6.3 Protocol

Report Form Issuing Sub-Protocol

The Report Form Issuing sub-protocol that is depicted in Figure 6.9 is run when the user needs to acquire a new Report Form.

1. The user generates a symmetric key $K_{user}$ and composes it into an RF-Request which is encrypted with the Broker’s public key $BroK^{Pub}$.

2. The RF-Request is sent to an Agent with the intent to be delivered to an Ad-Dealer.

3. The Agent transfers the RF-Request to an Ad-Dealer.

4. The Ad-Dealer forwards the RF-Request to the Broker.

5. The Broker decrypts the RF-Request with his private key $BroK^{Pri}$ and obtains the user’s $K_{user}$. The Broker then issues a new ARC-ID in the SC-Board and afterwards computes a pair of asymmetric keys $RepK^{sig}_{user}$ and $RepK^{ver}_{user}$. The Verification Key $RepK^{ver}_{user}$ is stored securely while the Signing Key $RepK^{sig}_{user}$ and the new ARC-ID are composed into a Report Form which is encrypted with $K_{user}$.

6. The encrypted Report Form is sent back to the Ad-Dealer.

7. The issuing Ad-Dealer verifies the transaction by signing the appropriate field on the SC-Board.
8. The Ad-Dealer forwards the Report Form to the Agent.

9. The Agent transfers the Report Form back to the user.

10. The user receives the encrypted Report Form and decrypts it with his copy of $K_{user}$ in order to obtain the ARC-ID and $RepK_{user}^{sig}$.

Figure 6.9: Report Form Collection sub-protocol.

**Ad-Report Submission Sub-Protocol**

The Ad-Report Submission sub-protocol that is depicted in Figure 6.10 is used for the delivery of Ad-Reports by the user to the Broker.

1. The user gradually composes an ARC. The contents of the ARC are encrypted with the Broker’s public key $BroK_{Pub}$ except for the first
CHAPTER 6. ADS+R: ADVERT FRAUD PREVENTION

block that contains the ARC-ID and the sequence number $N$ and hash digest $h(N−1)$ in all remaining blocks. When the ARC is ready for submission, the user produces an Integrity Hash $IH$ by computing $h(ARC)$ and signs the result with the signing key $RepK_{\text{user}}^\text{sig}$.

2. The ARC and IH are sent to an Agent with the intention to be forwarded to an Ad-Dealer.

3. The Agent transfers the ARC and IH to an Ad-Dealer.

4. The Ad-Dealer keeps a local copy of the ARC and IH.

5. The Ad-Dealer submits the ARC and IH to the Broker.

6. The Broker first decrypts the ARC with his private key $BroK^{\text{Priv}}$ and verifies the authenticity of the IH with the matching verification key $RepK_{\text{user}}^{\text{ver}}$. The Broker then verifies the integrity of the ARC by replicating the results of the hashes $h(N−1)$ in the individual blocks as well as the digest of $h(ARC)$ that is found in the IH. When verification has been completed successfully the Broker uploads the ARC and IH onto the SC-Board in plaintext form. Finally, the Broker verifies the validity of the Checkpoint Blocks CBs and Affiliation Blocks ABs and marks them on the SC-Board.

7. When the ARC and IH have been uploaded to the SC-Board, the Broker notifies the submitting Ad-Dealer with a Check Message.

8. The submitting Ad-Dealer verifies the hashes in the uploaded ARC and IH by comparing them to his own copy (recall that the hashes were not encrypted in the original ARC). The submitting Ad-Dealer then confirm the correctness of the ARC by placing his name and signature in the third field of the SC-Board.

9. The Broker notifies the Advertisers for the new submission in the SC-Board.

10. The Advertisers begin to reward Publishers and Ad-Dealers and each reward is marked on the SC-Board by the appropriate Advertiser. Each Advertiser is responsible for individually determining the honesty of the user by assessing the embedded authentication credentials which have been marked (CBs, ABs, RoCs and RoAs). The RoCs and RoAs are validated and marked on the SC-Board by the respective Advertisers after confirming the contained C-Tokens and A-Tokens.
Depending on the number and significance of credentials in the ARC, an Advertiser may choose to award a report or wait for more credentials to be marked on the SC-Board (more awarded RoCs and RoAs by other Advertisers and more confirmed ABs).

11. After a certain time has passed from the submission of the ARC, the user’s mobile client checks the SC-Board in order to determine that the ARC has been submitted. If the matching ARC-ID is present within the SC-Board, the user may discard the original copy of the ARC and IH or else she may need to resubmit them. Recall that the only part of the SC-Board which is visible to the user is the ARC-ID while the rest is kept private among the remaining stakeholders.

Figure 6.10: Ad-Report Submission sub-protocol.
6.4 Evaluation

To evaluate $ADS+R$ we follow a qualitative approach based on the system requirements which we determined in Section 6.1.3. We determine possible attack scenarios and gauge the level of threat which they pose to the system in terms of feasibility, practicality and likelihood.

6.4.1 Reporting Effectiveness

In the currently deployed system, Publishers need to rely on Ad-Networks in order to claim rewards from Advertisers. Although Ad-Networks have no financial benefit from mismanaging reports, the lack on transparency is a downside of the system’s reporting effectiveness. $ADS+R$ overcomes this limitation by offering Publishers the ability to audit the reporting process. As ARCs are published on the SC-Board, Publishers can claim their rewards directly from Advertisers without needing to trust any third parties.

In addition to transparency, our approach also offers supplementary reporting information which can be exploited for purposes of market research. To conduct market research, Advertisers currently rely on big data analysis which requires additional funding and in many cases can be intrusive for users. In contrast, $ADS+R$ maintains user privacy but at the same time provides fine-grained insight on consumer habits. By examining the contents of ARC, Advertisers can associate advertising strategies to social behaviours (e.g., consumers who are interested in product A are also interested in product B or consumers who visit location X tend to view adverts through Publisher Y). This renders ARCs as a more effective means of advert reporting both in terms of comprehensiveness and privacy.

6.4.2 Reporting Fraud Prevention

Financial fraud against Advertisers is the main shortcoming of the currently enforced model. The main perpetrators of fraud are BotNets (automated clickers) and Click-Farms where low-paid workers are hired to click on adverts. Such schemes commit fraud by generating a large bulk of Ad-Report traffic that does not correspond to actual consumer activity. To combat this problem, $ADS+R$ enables Advertisers to (1) calculate the rate upon which Ad-Reports are created and (2) verify that the user who submitter a particular Ad-Reports is an actual consumer rather a fraudster. The system’s effectiveness at detecting fraudsters is akin to the quantity and variety of verification credentials within each submitted ARC. A high concentration
of assorted credentials signifies user honesty however, a low concentration of credentials does not necessarily indicate dishonesty as it may also be attributed to a lack of activity on behalf of the user. In this regard, the behavioural verification mechanism of ADS+R may return a false positive when the credentials of an ARC are insufficient to yield a conclusive verdict. Additionally, a false negative may also be a theoretically possible if a fraudster were able to forge fake credentials. To overcome this limitation, ADS+R enables Advertisers to individually determine the validity of a submitted ARC based on statistical standards such as the average number of social affiliations, average rate of advert viewing and average rate of visitations of specific locations. The aforementioned statistical standards represent the typical behavioural patterns of honest users and can be set by analyzing previously sublimed ARCs that have been accepted by the Advertisers as real. Naturally, there is always the possibility that some previously accepted ARCs may have actually been false negatives. Despite this, provided that the sample of ARCs is large enough and considering the fact that our systems assumes that the majority of participants are honest, it is safe to infer that a limited number of ARCs which are false negatives will have minimal effect at skewing the results of the statistical analysis.

To illustrate the system’s ability to distinguish between honest and dishonest users, we will examine an attack scenario where a dishonest user $\hat{U}$ attempts to commit fraud at a large scale by submitting a set $\hat{S} = \{\hat{ARC}_1, ..., \hat{ARC}_i\}$ of fictitious ARCs. Recall that ARCs contain the following types of blocks: RoV (Report of View), RoC (Report of Click), RoA (Report of Action), CB (Checkpoint Block) and AB (Affiliation Block).

Among all types of Ad-Reports, RoV is the easiest to fabricate as it does not contain any verifiable information (Tokens). However, $\hat{U}$ cannot submit an $\hat{ARC}$ which only contains RoVs as this would be immediately rejected by the Broker. RoCs and RoAs contain a C-Token (Click-Token) or an A-Token (Action-Token) which can only be obtained by visiting an Advertiser’s website within a particular time-frame. This prevents $\hat{U}$ from creating RoCs and RoAs ahead of time but $\hat{U}$ can still attempt to commit fraud by creating a $\hat{ARC}$ over a longer period of time. Although this makes the creation of the $\hat{ARC}$ more difficult, it is still plausible with the use of an automated process that automatically downloads Tokens when they become available. This operation is very elaborate and time-consuming and therefore impractical for use at a large scale. Despite this fact, the $\hat{ARC}$ would still be rejected by the Broker due to the lack of additional verifies such as CBs or ABs.
CBs contain a time-stamp and a hash digest of the last block on an ARC which have been signed with the issuing Ad-Dealer’s Token Singing Key $\text{ToK}_{\text{Ad}}^{\text{sig}}$. This makes it impossible for $\hat{U}$ to forge CBs or use genuine CBs on multiple fake ARCs. In order to obtain valid CBs, $\hat{U}$ has no other alternative than to repeatedly travel to the physical location of an Ad-Dealer throughout the course of the creation possess of the fictitious ARC. Moreover, $\hat{U}$ would need to be cautious not to request multiple CBs (for different ARCs) at the same time as this would provoke suspicion. Even if $\hat{U}$ were to conspire with one of the Ad-Dealers, a fake ARC would still be in danger of being exposed due to the disproportionate number of CBs from just one source. For such an attack to be successful, $\hat{U}$ would need to conspire with multiple Ad-Dealers and manage CBs in such a way that does not create an observable pattern (e.g., multiple ARCs containing CBs from the same group of Ad-Dealers). Considering the fact that Ad-Dealers run the risk of being exposed, it would be unlike for $\hat{U}$ to be able to secure the cooperation of a large enough number of compromised Ad-Dealers.

ABs are exchanged between users and serve a similar purpose as CBs as they can be used to determine the rate in which the Ad-Reports of an ARC were created. In contrast to CBs however, the T-Token which is contained in ABs is not signed by an Ad-Dealer but by another user. This makes ABs vulnerable to forgery as $\hat{U}$ can exchange ABs between multiple fake ARCs. However, if $\hat{U}$ were to compose ARCs in such a manner, suspicions would still be raised by the Broker due to the lack of CBs, RoCs and RoAs.

To conclude, in order for $\hat{U}$ to fabricate ARCs which are realistic enough to fool the Broker, $\hat{U}$ would need to use an automated process which downloads C-Tokens and A-Tokens over an extended period of time. During that time, $\hat{U}$ would need to exchange ABs between the ARCs and also physically collect CBs from different Ad-Dealers without raising their suspicion by submitting multiple requests at the same time. Click-farms lack the mobility, sophistication and practical ability to operate in such a manner while automated clickers and bot-nets are limited by the time restraints of the process which makes the conduct of large scale fraud impractical and ineffectual.

To further solidify the robustness of our verification method against improbable but theoretically plausible behavioural patterns and specialized attacks which have been tailored specifically for the purposes of bypassing ADS+R, will now examine a series of behavioural scenarios and assess their likelihood and practical feasibility.
**Honest user imitates dishonest behaviour:**  
*ADS+R* recognizes three types of characteristic behaviour which may be attributed to a dishonest user: (1) high-rate advert engagement, (2) lack of mobility and (3) lack of social interaction. Any display of the aforementioned behaviours by a honest user can potentially result in a false positive labeling by *ADS+R*. More specifically, a high-rate of advert engagement is a characteristic behaviour of auto-clickers, click-bots and click-farms. Auto-clickers typically generate hundreds of *Ad-Reports* per minute which is practically impossible for a human operator to achieve. Sophisticate click-bots and click-farms may potentially be tuned down to generate a reduced number of *Ad-Reports* per minute but even this would still be too great for a normal consumer if we consider the fact that mobile apps are restricted to the display of up to three adverts per minute. A lack of mobility and social interaction are typical behaviours of click-farms but may also be potentially be exhibited by a honest user who is physically restricted and remains isolated for extended periods of time. Such a radical lifestyle is not entirely typical for most individuals but theoretically possible. Considering the fact however that the *ADS+R* relies on the distributions of promotional material either from *Ad-Dealers* or *Agents*, it would be practically unfeasible for a user who adopts such a lifestyle to be able to obtain adverts. We can therefore conclusively state that any user who is lacking mobility and social interaction may have fewer behaviours which prove their honesty but at the same time would be unable to participate in *ADS+R*.

**Click-farm imitates honest behaviour:** The definitive characteristic of click-farms is the lack of mobility which renders them unable to collect *Checkpoint Blocks CBs* from designated *Ad-Dealers*. Furthermore, depending on their size and mode of operation, each mobile device of a click-farm produces *Ad-Reports* at a rate that may not be as high as that of auto-clickers but will still be substantially larger than what is expected from a normal consumer. In order for a click-farm to imitate honest behaviour, three operations would need to take place. Firstly, the devices within the click-farm would need to imitate social interaction which is theoretically possible by exchanging *Affiliation Blocks ABs*. Secondly, the operator of the click-farm would need to click on adverts and refresh the ad banners on each of the devices at a rate that more closely resembles the activity of a genuine consumer. The success of such a mode of operation would call for a great concentration of a fraudulent devices which would only be used scarcely and for short intervals of time. Not operating at its fullest capacity,
would result in diminished profits for the click-farm which would likely make it unprofitable when considering the prices of smart phones and the power which is required to run them. Lastly, the click-farm would need to fake mobility by embedding the ARCs of each fraudulent device with CBs. Considering the fact that CBs are only available at the designated locations of Ad-Dealers, the fraudster would be required to fist build some from of custom hardware device which remotely relays CBs and then manually carry this device within range of various Ad-Dealers while at the same time being mindful of the rate of requested CBs as to not raise suspicions. Although hypothetically plausible, carrying out such en elaborate scam requires sophisticated technical knowledge and great deal of physical effort on behalf of the fraudster.

**Click-bot imitates honest behaviour:** Elaborate click-bots are known to imitate user behaviour by performing complex online activities such as navigating through websites, sending emails and even logging in fake social media accounts. Considering the fact that click-bots victimize inspecting users, an infected device within ADS+R could be committing fraud by embedding fake Ad-Reports within a user’s ARC which also contains legitimately acquired CBs and ABs. Under such circumstances, the user’s mobility and social interactions would play no significant role in the verification process but the click-bot could still be detected due to the high rate of generated Ad-Reports. Click-bots rely on the fact that traditional fraud detection systems have limited memory and processing power which restricts them from easily detecting Ad-Reports which originate from the same source. However, ADS+R composes all the Ad-Reports which are generated by the same user into an ARC which makes it impossible for a click-bot to commit fraud without being detected. In order for a click-bot to successfully replicate the behavioural patterns of a honest user, the rate of generated Ad-Reports would need to be reduced at a more realistic level which would also reduce the revenue of the fraud. Imitating honest behaviour is therefore possible for a click-bot but it also has limited profitability against ADS+R.

### 6.4.3 Reporting Integrity

After an ARC leaves the user’s device, it has to go through an Agent, an Ad-Dealer and the Broker before finally being posted on the SC-Board (Service Confirmation Board). This makes it possible for any of the intermediaries to commit fraud by altering the content of a ARC. This type of fraud would be particularly difficult to detect due to the fact that a legitimate ARC (one
created by a real user) is likely to have valid Tokens. ADS+R prevents this attack through the employment of hash functions and verification checks. To demonstrate the operation of the integrity mechanism, we will consider two attack scenarios.

**Attack scenario 1:** The Agent and the submitting Ad-Dealer attempt to alter the content of a legitimate ARC in order to trick the Broker and Advertisers into rewarding a malicious Publisher for a publication that did not take place. Recall that the ARC follows the architecture of a blockchain where the first block holds a unique ARC-ID and each following block $N$ includes the hash digest $h(N-1)$ of the previous block. Additionally, the user also sends an IH (Integrity Hash) that contains the hash digest of the entire ARC which has been signed with a Report Signing Key $RepK_{\text{sig}}^\text{user}$.

Since the content of the Ad-Reports is encrypted with the Broker’s public key $BroK_{\text{Pub}}$, it would be possible for a malicious Agent or Ad-Dealer to create a fictitious Ad-Report. However, if the fictitious Ad-Report were to be inserted into the ARC (either as a new block or by replacing an existing one), this would result in a mismatch of both the hash digests within ARC’s blocks ($h(N - 1)$) as well as the hash digest that is included in the IH ($h(ARC)$). The hashes $h(N - 1)$ within each block are in plain-text and therefore an attacker could be able to change them but the $h(ARC)$ within the IH is signed and can therefore not be altered without the user’s Report Signing Key $RepK_{\text{sig}}^\text{user}$. To obtain $RepK_{\text{sig}}^\text{user}$, the attacker would need to intercept the user’s RF-Request (Report Form Request) or RF (Report Form) which is not possible without access to the Broker’s private key $BroK_{\text{Pri}}$.

**Attack scenario 2:** The Broker attempts to alter the content of a legitimate ARC in order to cheat a Publisher or Ad-Dealer out of a reward. ARCs are encrypted by users with the Broker’s public key $BroK_{\text{Pub}}$ and are published on the SC-Board (Service Confirmation Board) only after they have been decrypted with the Broker’s private key $BroK_{\text{Pri}}$. The Broker could therefore attempt to cheat Ad-Dealers and Publishers by altering the Ad-Reports of a submitted ARC before uploading it to the SC-Board. To prevent this attack, the submitting Ad-Dealer (the Ad-Dealer who forwards the ARC to the Broker) holds a copy of the ARC in order to certify the Broker’s integrity. Although certain parts of the ARC are encrypted with Broker’s public key $BroK_{\text{Pub}}$, the hash digests $h(N - 1)$ are transferred in plain-text and are therefore legible to the submitting Ad-Dealer. This allows the submitting Ad-Dealer to replicate the hash functions on the posted
(decrypted) ARC in order to verify that decryption has been completed correctly. The submitting Ad-Dealer then marks the verification check VC-S in the SC-Board which informs the Publishers and remaining Ad-Dealers that the submitted ARC is valid.

6.4.4 User Privacy

ADS+R maintains user privacy on Ad-Reports via the same means as ADS does on adverts. ARCs are encrypted with the Broker’s public key BroK^Rpub which ensures user privacy against Agents and Ad-Dealers. The use of Agents (as partially trusted proxies) and the incorporation of anonymous-download protocols also provides additional layers of privacy against Ad-Dealers and the Broker. The same anonymous-download protocols are also used for the collection of CBs (Checkpoint Blocks). Furthermore the requesting as well as the collection of CBs is done over encrypted channels to ensure privacy against nearby eavesdroppers. ABs (Affiliation Blocks) reveal meetings between users but do not expose their identities to the Broker. The users who participate in an AB exchange only swap the hash digests of the last block within their respective ARCs and therefore have no means of obtaining any information about each others Ad-Reports. At any given moment, the Broker and Ad-Dealers have no means of obtaining the user’s identity while Agents and other system users (who can be assumed to already know each others identities) have no access to Ad-Reports.

6.5 ADS+R: Advert Fraud Prevention Summary

In this chapter we address the problem of advertising fraud and present ADS+R as a potential solution. ADS+R was implemented based on the infrastructure of ADS with the inclusion of additional elements and technologies. The main feature of ADS+R is the ability to identify fabricated Ad-Reports without compromising the privacy of the submitting users. To attain this goal, ADS+R features an innovative verification method which allows Advertisers to determine the honesty of users based on their behavioural patterns. Our qualitative evaluation indicates that ADS+R offers substation user privacy, protection against all common types of fraudsters, protection against compromised system actors who may alter the content of legitimate Ad-Reports and robustness against specialized attackers who explicitly aim to bypass the system’s security by impersonating honest behaviour.
Chapter 7

Conclusion

In this thesis we performed a thorough examination of the OBA (Online Behavioural Advertising) ecosystem and called attention to the threats it poses for user privacy and Advertiser security against fraud. After analyzing the state of the art, we reached the verdict that previous attempts to address user privacy and fraud prevention as separate issues have been able to partially resolve one of the two problems, but only at the expense of the other. Consequently, the currently available privacy-preserving advertising systems are susceptible to fraud or fail to offer fine-grain targeting, making them undesirable by Advertisers, while the systems that focus on fraud prevention require the collection of private data which renders them a threat for users.

After considering all of the parameters, we presented ADS+R as an innovative advertising system which supports the delivery of private and personalized adverts as well as the submission of verifiable anonymous Ad-Reports. To the best of our knowledge ADS+R is the first advertising system to achieve user privacy as well as fraud prevention, effectively underlining the significance of our research. Our qualitative analysis showed that ADS+R offers the following advantages in comparison to analogous systems:

Increased user privacy against other parties: ADS+R incorporates a fusion of multiple privacy-preserving mechanisms such as decentralized networking, peer-to-peer (proxy) connections and anonymous-downloading. Subsequently, ADS+R offers increased privacy in comparison to alternative systems which typically present a single point of failure.
User privacy against other users: Previous designs which benefit from social networking generally assume users as trusted. ADS+R avoids such assumption and treats users as malicious, preventing them from obtaining private data through the application of public-key cryptography and privacy-preserving protocols (used for performing user profile comparisons).

Robustness against sabotage: Alternative systems frequently focus exclusively on privacy or fraud prevention while ignoring security against sabotage such as impersonation attacks or data injection. ADS+R prevents such attacks by integrating strong authentication protocols, public-key cryptography and cryptographic hash functions.

Anonymous fraud prevention: ADS+R was designed with the intent to actively preserve user anonymity, while in most of the previous attempts to combat advertising fraud anonymity was considered to be outside the research scope.

Reporting integrity: In contrast to the currently adopted OBA system, ADS+R allows Advertisers and Publishers to directly audit the integrity of Ad-Reports thus preventing the integrity of submitted Ad-Reports from being called into question. Beyond the improvements in fields of advertising privacy and fraud prevention, ADS+R has also been shown to offer additional functional benefits which are listed as follows:

Resource conservation: ADS+R encompasses a mechanism which allows users to identify shared advertising interests and collectively have access to the same encrypted adverts while still preserving their privacy. The aforementioned aspect of ADS+R is not restricted to advertising systems but can also be used in other applications which make use of peer-to-peer networking.

Consumer data collection: A supplementary characteristic of ADS+R is the ability to collect consumer behaviour data. Set feature is not a vital trait of advertising systems but rather serves the complementary function of assisting in market research. It needs to be stated that ADS+R performs this operation without compromising user privacy.
CHAPTER 7. CONCLUSION

7.1 System Limitations

$ADS+R$ relies on user participation to propagate data, much like analogous systems which make use of social networking such as [70, 71, 133, 119, 111, 58, 57, 14, 125] and [147]. Subsequently, the performance of $ADS+R$ is relative to the number of users who participate in the system. At the worst case scenario where participation is minimal, the performance of $ADS+R$ at distributing adverts will demote to a similar level as contemporary anonymous-download schemes such as [124, 23, 24, 70, 71] and [133]. More specifically, even if no Agents were available, users would still be able to directly collect adverts from Ad-Dealers. However, users would need to rely on their own mobility to physically commute to the designated locations where set Ad-Dealers are broadcasting. In regards to fraud prevention, users would still be able to compose ARCs ($Ad$-$Report$ $Chain$) but without the inclusion of $AB$s ($Affiliation$ $Blocks$)\(^1\). Depending on the number of the remaining available verifiable blocks of an $ARC$ ($RoC$, $RoA$ and $CB$), the Broker would may still be able to authenticate the legitimacy of a submission but with less confidence. As possible ways to promote user participation, we propose the following ideal:

- **Promoting user privacy**: User participation can be encouraged by promoting $ADS+R$ as a more private alternative to $OBA$ and by calling attentions to the significance of advertising privacy.

- **Providing financial incentives**: Financial incentives can be provided for participants in the form of reward points and exclusive coupons. Financial incentives have also been proposed in previous attempts such as [133, 119, 111] and [146].

- **Integrating $ADS+R$ in mobile devices**: As a last resort, user participation could be enforced by integrating $ADS+R$ in mobile devices. Although drastic, such as measure is realistically possible if promoted by Advertisers who also benefit from the system.

A second limitation of $ADS+R$ is in regards to the use of $Tokens$. The security of $ADS+R$ hinges on the principle that the Broker has access to $Tokens$ but remains unaware of the user’s identity. If anonymity were to be compromised, the Broker would not only uncover the user’s advertising interests but also additional information such as browsing habits, social affiliations and location patterns. A compromise of the user’s anonymity

---

\(^1\) $AB$s are authentication $Tokens$ which are signed by other users.
could be achieved if the Broker were to tamper with the software client or collude with a malicious Agent. Although theoretically possible, such actions constitute as malicious behaviour on behalf of the Broker and therefore fall outside the scope of this research.

7.2 Practical Implementation

To evaluate the feasibility of our system within the context of the marketing industry, we perform a detailed examination of the conditions that need to be met in order for a practical implementation of ADS+R to be successful. Based on our assessment, the first factor which is essential for the implementation of ADS+R is the acquisition of Advertisers who are willing to participate in the system. Advertisers provide the monetary capital which drives the entire advertising industry. Under the currently enforced OBA system, Advertisers benefit greatly from the provision of adverts which are personalized to the particular advertising needs of the concerned consumers. However, the operation of the OBA system has gradually been changing over the past years due to a global shift towards the enforcement of user privacy regulations [73]. A prime example would be the enforcement of the General Data Protection Regulation (GDPR) which was established by the European Union in 2016 and became active in 2018. The GDPR dictates that all businesses within the EU are obliged to enforce strict privacy-preserving practices and are also prevented from processing personal data without first acquiring the explicit consent of the concerned users [144].

Research suggests that as the advertising industry adopts evermore strict privacy regulations such as the GDPR, the imposed cost for the Advertisers will have significant negative effects for small and medium companies who will find themselves being unable to compete with large firms [22]. Additionally, the lack of user identification data will also prevent analytics companies from easily identifying invalid traffic. As a result, a spike in advertising fraud may be caused which will add to the already significant problem and result in even greater monetary losses for Advertisers. It is, therefore, understandable that Advertisers who may be concerned about the added cost and progressive decline of OBA’s targeting effectiveness will start seeking a suitable alternative system.

ADS+R was explicitly designed to offer both user privacy and fraud resilience which it the ideal alternative to OBA. The client-side targeting function which is incorporated into ADS+R ensures that private data never leaves the user’s device. ADS+R is therefore unaffected by any privacy reg-
ulation that may be imposed on the advertising market. At the same time, \( ADS+R \) offers fine-grained targeting capability which may even be superior to the currently adopted \( OBA \) model and also protects against fraud. The combination of fine-grained targeting, fraud prevention and compliance to user privacy standards constitute as notable incentives for \( Advertisers \) to invest in \( ADS+R \).

One limiting aspect of \( ADS+R \) which needs to be considered however, is the lack of a central authority which would make the coordination of \( Advertisers \) more difficult. To overcome this limitations, we propose the creation of a self-regulatory federation of \( Advertisers \). The main role of self-regulatory bodies is to establish a set of standards that all members need to adhere to. Establishing a federation of this nature for the needs of \( ADS+R \) should be easy to achieve considering the fact that analogous self-regulatory organizations are already in operation with the most notable example being \( AdChoises \). \( AdChoises \) is a self-regulatory program which call for \( Advertisers \) to enforce targeting practices that comply with the privacy needs of users [155]. Similarly to the way that \( AdChoises \) operates, a self-regulatory organization of elected \( Advertisers \) can be formed for the purpose of supervising the operation of \( ADS+R \). This organization will be given the responsibility for managing the \( Broker \) and \( Ad-Dealers \) and will be tasked with performing administrative tasks such as employing the personnel that operates as the \( Broker \), maintaining the system’s infrastructure as well as selecting and managing the \( Ad-Dealers \).

The last factor that affects the practical feasibility of \( ADS+R \) is the development and maintenance of the hardware and software infrastructure that the system is based on. The two main elements of infrastructure which are required by \( ADS+R \) are: (1) the mobile client software which operates on the user side and (2) the networking devices which are hosted by the \( Ad-Dealers \). The user client is responsible for composing the user’s \( Advertising Interest Profile (AIP) \), adding reports to the user’s \( ARC \) and exchanging data with \( Ad-Dealers \) and other users. The functions of the mobile client can be performed by a simple smart-phone application which has been given access to the sensors and memory of the user’s device. Developing, distributing and maintaining a software application capable of performing the set tasks is relatively easy and inexpensive. The networking device on the \( Ad-Dealer \) side serves as an anonymous communication gateway between \( Advertisers \) and the mobile clients of users who appear within range. The \( Ad-Dealer \)’s networking device functions much like a WiFi access point with the sole difference that it does not make use of standard networking protocols in order to maintain the anonymity of the connection. In this regard,
ADS+R does not require the development of any custom hardware but can instead operate with the use of modified firmware which is installed on the routers of the Ad-Dealers. Modifying the firmware of a router in order to accommodate anonymous connections is relatively easy and has already been implemented by several analogous systems such as [124, 23, 24, 70, 71] and [133]. However, performing such a modification on multiple different router devices and establishing compatibility between them may be impractical. To work around the problem, instead of updating the existing infrastructure, a supplementary router device, that will be used exclusively for ADS+R, could be offered to the Ad-Dealers. Having a dedicated router will increase the initial cost of the system but will also simplify the setup process and reduce the long term maintenance cost.

All things considered, we can infer that an implementation of ADS+R is practically attainable in terms of technological development as well as in terms of functional integration within the currently enforced digital advertising model.

### 7.3 Future Work

ADS+R offers a great deal of versatility which allows for the incorporation of supplementary functionality. Some of the features which may be added to ADS+R are listed in the following paragraphs.

**Additional Tokens** The current interactions of ADS+R only utilize four types of authentication Tokens, but the system can easily accommodate more Tokens based on additional user behaviours. Some examples of common user behaviours are listed as follows:

- Offline Payment: A Token is collected from a physical retailer when the mobile device is used to perform an offline payment.

- IoT proximity: A Token is collected when the device enters the proximity of IoT devices (for e.g., proximity beacons, virtual assistants, remotely connected devices).

- Sensor Data: A Token is generated by the device itself when certain environmental conditions are detected by the on-board sensors (for e.g., the phone is moved or inserted onto a pocket).
• Online Activity: A Token is collected when the device performs certain online activities such as accessing social media sites or utilizing messages services.

Incentive Mechanisms As discussed in Section 7.1, ADS+R could benefit from the incorporation of an incentive mechanism which would reward Agents for offering their services to other users. Such a mechanism could be implemented in the form of reward points which would serve the role of a digital currency within the system. It needs to be noted that in order to comply with the overall design of ADS+R, the set incentive mechanism would need to preserve user privacy.

7.4 Final Remarks

Despite the rapid development of the online advertising industry within the past decades, the mechanism which are currently used for advert targeting and fraud prevention are seriously lacking in terms of privacy and security. This state of affairs can be attributed to the fact that digital advertising corporations are mainly focused at maximizing their profitability while disregarding the interests of users and business. All previous attempts by academics to resolve the pressing issues of digital marketing have mostly been unsuccessful. Part of the accountability for this lack of success on behalf of the research community can be ascribed to the fact that there is a serious deficiency of resources which greatly perplexes the development of new systems. To effectively conduct their research, scholars require access to information such as the algorithms which are used for matching behavioral triggers to specific advertising interests, accurate trace-sets of consumer mobility patterns and detailed reports of previously detected instances of advertising fraud. However, due to lack of cooperation on behalf of the advertising firms and limited funding, such useful insight is not always accessible and researchers are often forced to developed their systems based on approximated parameters and simulated data-sets. Similar challenges were faced during the conduct of this research. Due to the scarcity of suitable data-sets which accurately document the advertising interests and social interactions of users, we had to adapt and modify an akin data-set of Foursquare venue check-ins. Regardless of the setbacks that were faced, the successful completion of ADS+R marks a milestone in the development of security oriented advertising systems and forms a solid base for further research.
Bibliography


BIBLIOGRAPHY


