future internet

Article

Private and Secure Distribution of Targeted Advertisements to Mobile Phones

Stylianos S. Mamais and George Theodorakopoulos *

School of Computer Science and Informatics, Cardiff University, 5 The Parade, Roath, Cardiff CF24 3AA, U.K.; MamaisSS@cardiff.ac.uk

* Correspondence: TheodorakopoulosG@cardiff.ac.uk Tel.: +44-29-208-74855

Academic Editor: Georgios Kambourakis
Received: 4 April 2017; Accepted: 26 April 2017; Published: 1 May 2017

Abstract: Online Behavioural Advertising (OBA) enables promotion companies to effectively target users with ads that best satisfy their purchasing needs. This is highly beneficial for both vendors and publishers who are the owners of the advertising platforms, such as websites and app developers, but at the same time creates a serious privacy threat for users who expose their consumer interests. In this paper, we categorize the available ad-distribution methods and identify their limitations in terms of security, privacy, targeting effectiveness and practicality. We contribute our own system, which utilizes opportunistic networking in order to distribute targeted adverts within a social network. We improve upon previous work by eliminating the need for trust among the users (network nodes) while at the same time achieving low memory and bandwidth overhead, which are inherent problems of many opportunistic networks. Our protocol accomplishes this by identifying similarities between the consumer interests of users and then allows them to share access to the same adverts, which need to be downloaded only once. Although the same ads may be viewed by multiple users, privacy is preserved as the users do not learn each other's advertising interests. An additional contribution is that malicious users cannot alter the ads in order to spread malicious content, and also, they cannot launch impersonation attacks.

Keywords: online behavioural advertising; opportunistic networks; mobile networks; privacy; privacy-preserving advertising

1. Introduction

Advertising has played a very significant role in the expansion of the digital media industry over the past few years. The ability to generate revenue through the publishing of adverts allows companies to offer a wide range of services without any cost for users while still making profit.

Although digital advertising has been around for almost as long as the Internet itself, the latest financial studies suggest that the advertising market has experienced a rapid increase in recent years with a prime focus given to adverts that are specifically designed to target mobile devices. This shift towards mobile ads is driven by feedback from the online browsing habits of the consumers themselves [1]. Research analysis indicates that on average, an adult will spend more than 2.5 h a day on his/her smartphone [1]. In the U.K. alone, spending on mobile ads has increased by 45% in the year 2015. This accounts for an expenditure of 3.2 billion pounds, which is predicted to further increase by another 35% in the following years [2]. With such investments on mobile advertising, it should come as no surprise that a great deal of effort is focused towards effective advert targeting.

Online Behavioural Advertising (OBA) is a highly effective method for distributing content-aware adverts that target consumers based on their individual interests [3]. From a business point of view, OBA is beneficial to both vendors and consumers as the former increase revenue and the latter are only presented with ads that are relevant to their needs. Regardless of the commercial benefits, targeted advertising raises serious privacy concerns.

1.1. Problem Statement

Targeting of content-aware adverts is only possible after analysing a user's behavioural conducts. This practice utilizes private data that may reveal sensitive information, such as demographics, ideological and religious inclinations, shopping and browsing practices, lifestyle preferences and pretty much any other intelligence that can be deduced depending on the platform.

Although OBA has been employed for years, the ever-increasing usage of smartphones further escalates the existing problem. In addition to being more pervasive than computers, mobile devices incorporate a multitude of tracking capabilities that computers simply do not have. Smartphones have access to information such as personal text messages, emails, GPS coordinates, WiFi access point locations, purchasing habits through applications, such as Apple pay or loyalty point apps, and activity sensors.

Although the general public may not be fully aware of the technical details of OBA, numerous relevant studies indicate that the majority of people disapprove of being tracked. Specifically, a survey conducted by Pew Research Center showed that 68% of Americans view targeted ads in a negative way [4]. Despite more than half of the population disapproving of ads, we still need to acknowledge that targeted ads are vital for the offering of free services. Finding an alternative method of delivering personalized ads without compromising privacy is therefore an important problem that deserves to be addressed.

1.2. Contribution and Paper Layout

In this article, we aim to address the privacy concerns of targeted advertising by offering an alternative ad-distribution system. The contribution of our system compared to existing approaches is that it eliminates the need for trust among users, reduces memory and bandwidth overhead and also addresses the issue of security against impersonation and fake-ad injection attacks. We begin by illustrating the operation of the current OBA model and indicating its issues in Section 2. In Section 3, we examine and classify other proposed ad-distribution systems and then evaluate them in terms of privacy, security, targeting effectiveness and practicality. In Section 4, we first consider our observations from the evaluation of the relevant work and then determine a threat model, as well as a set of operational requirements, which will serve as evaluation criteria for our system, which we present in Section 5. In Section 6, we evaluate our system with respect to the relevant work by examining different threat scenarios, which are determined based on the threat model and evaluation criteria that we define in the previous section. Finally, in Section 7, we summarize our findings and present our conclusions.

2. The OBA Model

Online Behavioural Advertising (OBA) is defined by the United States Federal Trade Commission (FTC) as "the practice of tracking an individual's online activities in order to deliver advertising that is tailored to the individual's interests" [5].

The OBA system that is currently operational consists of four components. The central component is known as the Ad-Networkand is responsible for monitoring the users' behaviour in order to present them with relevant ads. Directly connected to the Ad-Network are the Advertisers and the Publishers. Advertisers operate as representatives of businesses who wish to promote products and services through advertising campaigns. The Advertisers create promotional material and supply it to the Ad-Network along with pricing details, i.e., along with the price they are willing to pay to the

Ad-Network every time their ads are being clicked on or viewed. The Ad-Network is responsible for delivering the ads to the consumers.

The adverts reach consumers by being displayed on platforms that are commonly viewed by users. These platforms are known as Publishers and are primarily websites, software applications or other popular services that are regularly visited by users. For an agreed price, Publishers give full control of certain visual areas within their websites or software to the Ad-Network. These areas are known as Ad-Boxes and they are used to display adverts. When a user visits a website or service of a Publisher, the Ad-Network chooses one of the adverts that have been supplied by the Advertisers and then displays it within the Ad-Box.

3. Related Work

Increasing the privacy in targeted advert delivery models is a research topic that several academics have attempted to resolve. Researchers seem to 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.

In the following sections, we offer insight into the available advert delivery systems. We first classify them into three categories based on the approaches and methods that they use, and then, we provide an overview for some of the most notable systems for each category. Finally, we conduct a critical evaluation for each category and proceed to recognize their flaws and limitations.

3.1. Classification of Available Systems

Previously-proposed advert delivery systems incorporate various combinations of architectures, as well as privacy mechanisms and can be arranged in three general categories.

- Trusted proxy-based anonymity.
- Selection from pool of adverts.
- Anonymous direct download.

3.1.1. Trusted Proxy-Based Anonymity

The simplest method of achieving anonymity is by introducing some form of trusted third party that acts as a proxy between the user and Ad-Network. The role of the proxy is to mask the identity of the user by forwarding his/her 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 the requests, but cannot see the content of the requests nor the corresponding ads. In turn, the Ad-Network or any other entity that distributes ads can decrypt the encrypted requests, but is not aware of the user's true identity, as it is masked by the proxy.

3.1.2. Selection from Pool of Adverts

Schemes that are based on this approach make use of client-side processing by allowing users to select ads that best satisfy their advertising needs, out of a pool of ads. The pool can be populated by various methods with the simplest one being by making a generic request. When following this approach, the user issues a request for ads that fall under a very broad category, which includes his/her specific interest. For example, if the user is interested in running footwear, he/she may make a generic request for sporting equipment. The Ad-Network responds with multiple ads that satisfy the request, and it is up to the user to keep ads that best match his/her particular interest and discard the rest.

3.1.3. Anonymous Direct Download

A further group of proposed approaches directly download freely broadcast adverts through the use of specialized hardware and software. Advertisers store their ads at broadcasting stations that operate in publicly-accessible locations. As a user comes into proximity of these stations, his/her device downloads the available adverts. The user's device is then responsible for sorting through the collected ads 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 of any information that exposes the user's identity, such as username, network address or physical address. Some of these systems also enable the users to connect and exchange adverts with each other. In these systems, a user downloads ads from a broadcasting station and then he/she propagates them to other users that he/she later comes into proximity with. This extends the reach of the broadcasting stations, but it also requires a certain level of trust among the users.

3.2. Literature Review

The P2PMM-system [6] relies on a trusted proxy that is referred to as the Intermediary Services Provider (ISP). The ISP is entrusted to store the sensitive information of the user and directly answer requests with adverts that it has obtained from merchants. Although the ISP has no immediate interest to expose any information to the merchants, this method assumes that the ISP can be fully trusted. Obliviad [7] accomplishes a similar goal by replacing the proxy with a hardware device that is placed on the Ad-Network side. The device receives the requests from the client and then sends a number of matching ads 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 [8]. 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.

Adnostic [9] composes a local interest profile that does not get disclosed to other parties. When the user visits a website, he/she is sent a number of ads that are relevant to the contextual theme of that particular website. For example, if the user visits a travel website, the ads can be for holidays to various destinations. The ad that is the most relevant according to the user's interest profile is then selected and displayed, while the rest are discarded. Although this method may be considered secure, it produces unnecessary overhead. It can also be argued that targeting is not very effective, since the ads that are sent to the user are based on a very general interest assumption. Kodialam et al. [10] follow a similar approach as [9]. They propose a role reversal scheme where the ad providers send to the user a series of interest profiles along with a set of matching adverts. This approach may potentially be more effective than Adnostic [9], since the users have a wider variety of ads to choose from. However, the generated overhead due to unused ads still degrades efficiency. Privad [11] is based on the selection from an advert pool, but also incorporates a proxy. A trusted third party, the Dealer, operates between the user and the Ad-Network. The user selects a general interest category and sends it to the Ad-Network through the Dealer (proxy). Upon receiving the message, the Ad-Network uses the same path to respond with a wide variety of relevant ads. The user's device sorts through these ads and selects the most relevant to be displayed. Although this method is simple to incorporate into the existing model, it assumes the existence of a fully-trusted third party. The Dealer is also a single point of failure, and if compromised, the security of the entire system can be bypassed.

PervAd [12,13] provides personalized ads through broadcasting. Users who maintain a local interest profile can collect relevant ads as they move into proximity of customized WiFi access points. The system minimizes overhead by first sharing some contextual information about the available ads, thus allowing users to selectively download only specific content. The interest profile is specified by the user himself/herself, and the downloading process is performed anonymously. Even though this method achieves a substantial level of privacy, it is highly impractical as users need to physically travel to specific locations. The schemes in [14,15] also use broadcasting, but at the same time make use of opportunistic networking. In both schemes, the users download adverts directly from local businesses. As users come into proximity, their devices connect and exchange ads based on contextual information. A mechanism for keeping track and rewarding points for interactions is also established. Although this is necessary for identifying the users who contribute the most to the system by propagating ads to others, it may also be a threat to the privacy of the system as it exposes the identity of the users, as well as their social encounters. An additional factor that the system does not account for is the presence of malicious users that may affect the integrity of the system by spreading fake ads or malware.

The Let's Meet! [16] 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.

3.3. Classification Assessment

Our assessment of the related work is shown in Table 1. The previously-proposed systems still have certain shortcomings, which we will now analyse in more detail.

Systems that make use of a proxy can be easily incorporated into the current architecture and achieve an adequate level of privacy, without seriously reducing the effectiveness and the efficiency of the system. Regardless, such models assume the existence of a trusted or partially trusted third party that can act as the proxy. This is not entirely realistic and also creates a single point of failure that threatens the integrity of the system if compromised.

Systems that are based on the selection from a pool of adverts, maintain privacy by taking advantage of the computational capability of mobile devices. These systems 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 and resource efficiency. The information that is shared with the Ad-Network may be too generic to effectively retrieve adverts that perfectly correspond to the user's interests. A significant amount of overhead is also generated. This is due to the fact that the advert selection needs to be performed locally and also additional ads need to be downloaded and stored only to be discarded afterwards.

Table 1. Assessment of categories of ad-delivery systems (good: "+"; adequate: "×"; poor: "−").

	Trusted Proxy	Pool of Ads	Direct Download
Privacy vs. Ad-Network	×	×	+
Privacy vs. other users			−
Security vs. attacker	+	+	−
Targeting effectiveness	+	−	+
Practicality/usability	+	−	−
Resource conservation	+	−	×

Models that allow users to directly download content from broadcasting stations offer the highest level of security so far. In some of these systems, the user does not even need to share any information, and therefore, privacy is only dependent on the capability of mobile devices to connect to stations anonymously. These systems require the use of specialized equipment that may be costly or impractical to incorporate. Their effectiveness to deliver accurate content depends on the amount of information that the user shares with the station, the availability of relevant adverts and the user's ability to physically travel to the specified locations. Resource efficiency is also limited, as these models can be very demanding in terms of storage, processing power and battery life. When enhanced with opportunistic networking, the effectiveness of the system may increase, but this will also raise trust issues among the users. Users need strong incentives to actively participate in such systems and efficiency will be reduced even further as additional strain will be put on the mobile devices.

4. Proposed System Overview

The system that we propose is based on the traditional anonymous download architecture and also incorporates opportunistic networking. Fine-grained targeted adverts are dispatched from broadcasting stations and then are propagated across the nodes (users) of a social network. Our system improves on previous attempts, as it eliminates the need for trust among users, reduces memory and bandwidth overhead and also addresses the issue of security against impersonation and fake-ad injection attacks.

Our system is designed to be highly versatile in terms of ad-targeting and opportunistic routing. The targeting algorithm, which is the mechanism that determines the user's advertising needs, can operate independently of the rest of the system for as long as it produces an output of a standard format. The opportunistic routing algorithm that we use in this paper follows a probabilistic approach, which is inspired by PRoPHET [17]. However, our model is also compatible with a wide range of other opportunistic networking schemes.

4.1. System Stakeholders

Our system consists of four basic stakeholders that operate independently from each other and in accordance with their own incentives and goals. Advertisers are advertising firms that wish to promote products through advertising much like in the OBA system. Advertisers commission the services of a Broker, which is a privately owned company that generates revenue by distributing ads to consumers. The Broker delivers the adverts through a network of distributors that are called Ad-Dealers. Ad-Dealers have a physical presence in publicly accessible areas and are able to deliver adverts upon request by establishing wireless connections with the mobile devices of users. Users represent potential consumers who own smartphone devices and are targeted by advertising companies through the services they use, such as websites and free mobile apps.

The Broker is responsible for the Ad-Dealers in his/her network and therefore needs to institute some form of trusted relationship by performing background checks and signing legal agreements. Ideal to play the role of the Ad-Dealers are local entities that have the capability to install and manage advert distribution hardware in publicly-accessible locations such as malls, shops and local WiFi hotspots. The Ad-Dealers conduct their own business independently but at the same time share resources and work cooperatively under the administration of the Broker.

All of the participants operate independently and the system can easily be expanded. The Broker can recruit new Ad-Dealers by authorizing the installation and use of the required infrastructure, while new users only need to download and register the client software on their devices.

4.2. Trust and Threat Model

Advertisers are assumed to fully trust the Broker to distribute their adverts, as well as to provide them with accurate billing information. We need to point out that this assumption is not entirely realistic as the Broker has a great interest to lie to the Advertisers in order to overcharge them. As the main focus of this research is the privacy of the user, this constitutes an entirely different problem that only concerns the Advertisers and the Brokers. This problem is also present in the current OBA model and can be addressed in the future in a manner that does not affect the rest of the system.

The Ad-Dealers are considered by the Broker as trustworthy enough not to compromise the integrity of the system by exposing secret cryptographic keys or tampering with the infrastructure. No trust is required on behalf of the Ad-Dealers towards the Broker as he/she has no means or interest to interfere with the Ad-Dealers' business.

The Ad-Dealers, as well as the Broker are considered to be honest enough to provide relevant adverts that are not fake or malicious, but at the same time they are curious and very determined to obtain access to private data. The user can therefore trust the provided material but is not willing to expose any information that can link his/her true identity (name, address, banking information) to any specific advertising preferences.

Despite being part of the same social circle, users do not fully trust each other. A user can easily be compromised and act maliciously by revealing private information or by propagating content that is fake and harmful. Compromised users are also considered as a threat by the Broker and Ad-Dealers, as they can cause downtime or undermine the quality of service by attacking the system.

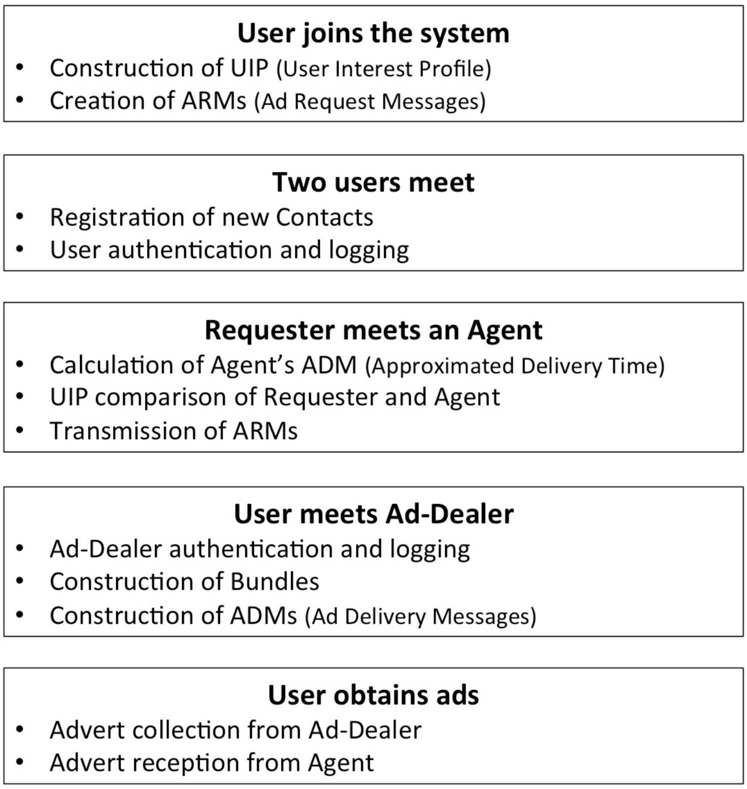
4.3. Evaluation Criteria

In consonance with the security threats that we recognize in the previous section and the shortcoming of analogous systems, we can now define a set of requirements that can serve as evaluation criteria for our system.

- User privacy against the Broker and Ad-Dealers: Neither the Broker nor his/her Ad-Dealers should be able to obtain any information that could be used to link a user's true identity to his/her advertising interests.
- User privacy against other users: Users should not have precise knowledge of the advertising interests of other users within their social circle.
- Protection from fake or harmful content: Attackers should not be able to infect the system with adverts that have not been distributed from a valid Ad-Dealer.
- Protection against impersonation attacks: Attackers should not be able to impersonate the identity of a user or an Ad-Dealer.
- Resource conservation: As mobile devices offer limited resources, the system needs to be conservative in the consumption of memory and battery power.

5. Detailed System Description

The Broker initiates the operation of the system by creating a grid of Ad-Dealers who are given access to specialized networking equipment and two public/private key pairs, one used for encryption/decryption and one for authentication. The Broker then develops the mobile client software and makes it accessible for users to download. Finally, the Broker accumulates ads from various Advertisers and forwards them to the Ad-Dealers. The ads are organized into groups based on their targeted audiences. When appearing at close proximity of an Ad-Dealer, the mobile client on a user's device can establish an anonymous connection and download ads for specific consumer interests on request. Additionally, users forward requests and adverts among themselves. When two mobile clients (users) A and B come into proximity, A sends ad requests to B. User B is responsible for collecting the relevant adverts from the Ad-Dealer by forwarding the requests and then delivering the ads back to A. The system comprises a sequence of sub-protocols that are triggered when certain events happen, e.g., when two users meet. We describe these sub-protocols and the corresponding triggers in Figure 1.



User joins the system

- Construction of UIP (User Interest Profile)
- Creation of ARMs (Ad Request Messages)

Two users meet

- Registration of new Contacts
- User authentication and logging

Requester meets an Agent

- Calculation of Agent's ADM (Approximated Delivery Time)
- UIP comparison of Requester and Agent
- Transmission of ARMs

User meets Ad-Dealer

- Ad-Dealer authentication and logging
- Construction of Bundles
- Construction of ADMs (Ad Delivery Messages)

User obtains ads

- Advert collection from Ad-Dealer
- Advert reception from Agent

Figure 1. System sub-protocols and their corresponding triggers.

5.1. User Joins the System

5.1.1. Construction of UIP

Smartphones have access to a multitude of user information such as browsing logs, emails, text messages, mobile app data, GPS coordinates and data from available sensors, such as accelerometers, pedometers and heart rate monitors. By simply tapping into these resources, the client can determine the users advertising needs with much higher effectiveness than a remote observer. For example, when the device detects high physical activity while in close proximity to a park, this can be associated to an advertising interest for sporting equipment.

The user's advertising interests are represented in the form of a list that follows a standard format throughout the entire system and is termed the User's Interest Profile (UIP). Each entry of the UIP denotes a common consumer interest (automotive, technology, food and drink, etc.) and is recognized by a unique identifier, which is noted as the Interest-ID. When given the right permissions, the client associates the user's activity with specific advertising interests that can in turn be marked on the UIP in a simple binary manner where "true" represents that the user's activity matches the particular interest of the UIP and "false" that it does not. The UIP is stored locally where it can be dynamically maintained in accordance to the alterations of the users behaviour and is not accessible to anyone else.

The determination of suitable interests happens independently from the rest of the system, thus offering a great deal of versatility. Developers will be able to fine-tune the system by designing their own algorithms that can be fully compatible with the rest of the system for as long as they produce an output which follows a standard UIP format.

5.1.2. Creation of ARMs

Once the UIP has been constructed, the marked interests that have been associated with the user's activity may be used to create requests for relevant adverts while the remaining entries are ignored. The client begins by generating a cryptographic key K_0^{user}, which is then used to create multiple one-time keys K_i^{user}, which are produced with the use of a hash chain:

$$K_i^{user} = h(K_{i-1}^{user}) \tag{1}$$

The client goes through the UIP and recovers the Interest-ID for each of the marked entries. For each of the marked interests, the client then creates an Ad Request Message (ARM), which contains the equivalent Interest-ID and a new K_i^{user}. The ARMs are then encrypted with a public key that comes pre-installed in the client and is known as the System Encryption Key (EK_{System}). Finally, the resulting encrypted ARMs, as well as a copy of each K_i^{user} are stored for future use. Note that this operation does not need to take place in real time. We can therefore significantly preserve resources by performing it while the device is idle and preferably connected to a power source.

$$ARM_{(ARM-ID)} = E_{EK_{System}}\left[interest - ID, K_i^{user}\right] \tag{2}$$

5.2. Two Users Meet

Mobile clients perform periodic scans (e.g., via Bluetooth or WiFi) to detect nearby Ad-Dealers and other clients. We need to 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.

5.2.1. Registration of New Contacts

When two mobile devices come into proximity for the first time, both clients request a manual confirmation from the users so that they can register each other as Contacts. Registration happens by trading a unique identity number that is built within each version of the client and will from now on be simply referred to as User-ID. An out-of-band channel (SMS, QR code, email or keyboard input) is then used to exchange a pair of authentication passwords, and the meeting is logged. Alternatively, the passwords could be exchanged automatically via Bluetooth. Although this would simplify the operation, it would also create the risk of passwords being sniffed by a nearby eavesdropper. The passwords are generated automatically by a secure random string generation function (more details on this are given in Section 6.

5.2.2. User Authentication and Logging

When two registered Contacts meet, their encounter is logged by both clients after the users have authenticated each other with a challenge-response password handshake.

As shown in Figure 2, the procedure begins with the two users A and B exchanging their User-IDs (U_A and U_B) and two random nonce challenges $R1$ and $R2$. User A hashes $R2$ concatenated with his/her password P_A and sends the result $H_A = h(P_A, R2)$ to User B. Similarly, User B produces $H_B = h(P_B, R1)$ and sends it to A.

Once H_A and H_B have been delivered, User A generates $H_B' = h(P_B', R1)$ where $P_B{}'$ is his/her originally stored copy of the password that B sent when they became Contacts, while User B performs the same operation to calculate $H_A' = h(P_A', R2)$. The two users can now authenticate each other by simply comparing the hash digests H_A with $H_A{}'$ and H_B with $H_B{}'$.

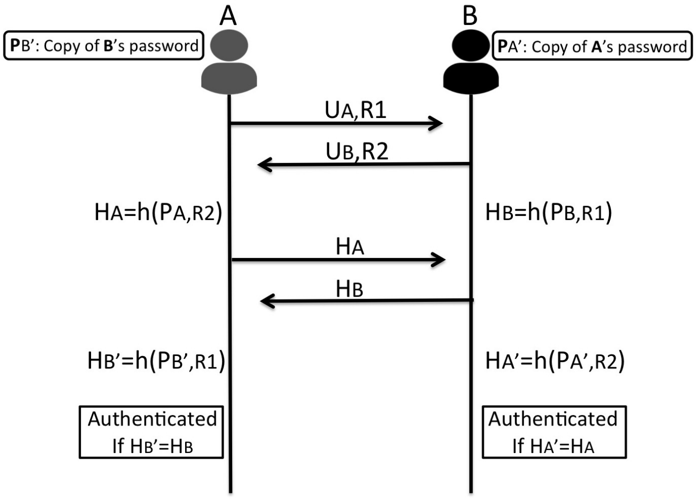


Figure 2. User handshake protocol for user authentication.

5.3. Requester Meets an Agent

As we explain in Section 4, adverts are propagated across the system through the use of opportunistic networking. Users download ads for their own use, but at the same time can collect ads on behalf of other users. This is highly beneficial for users that do not encounter Ad-Dealers often: such users can acquire adverts by exploiting the mobility of other users within the same social circle such as family members, friends and co-workers.

A user that ferries requests or adverts on behalf of another will from now on be called the Agent while the user who places the request will be called the Requester. The system can support Agents who ferry ads for multiple Requesters, and it can also support users who operate as both Requester and Agent (for another Requester) at the same time. However, for simplicity, we will examine the most basic scenario where two users meet and one of them plays the role of the Requester while the other serves as his/her Agent.

5.3.1. Calculation of Agent's DT

The opportunist network that we use operates as follows. A Requester sends ARMs to an Agent who forwards them to the Ad-Dealer when the two come within range. The Agent then receives from the Ad-Dealer ads that he/she conveys back to the Requester the next time they meet. For this scheme to be efficient, it is important that the Requester only uses the services of Agents who can deliver ads quicker than he/she would normally need to collect by himself/herself by visiting the Ad-Dealer.

The system adopts a history-based approach to determine the probability of the Requester receiving ads quicker if he/she were to use an Agent. This approach uses three metrics. The first metric is named Time To Ad-Dealer of a user (TTA_{user}), and it represents the approximate delay until the user is expected to appear again within the range of an Ad-Dealer. The TTA_{user} can be estimated by calculating the average time between successive encounters of the user with and the Ad-Dealers and subtracting from it the time that has passed since their last meeting (e.g., if the user encounters an Ad-Dealer every 24 h on average and it has been 4 h since the last encounter, then the TTA_{user} is $TTA_{user} = 24 - 4 = 20$ h).

The second metric is the Encounter Average $EA_{(u1,u2)}$ of two users, $u1$ and $u2$, and it denotes the average meeting time between successive encounters for a pair of users. It is obvious that since both users record the same meetings, $EA_{(u1,u2)}$ is expected to be equal to $EA_{(u2,u1)}$.

These two first metrics are calculated by both the Requester and the Agent with information that can be found in their own log files. The last metric is the Delivery Time of Agent (DT_{Agent}) and expresses the estimated period needed for a particular Agent to deliver an ARM to an Ad-Dealer and then convey back the requested advert. The DT_{Agent} is calculated only by the Requester, but also requires input from the Agent. The Requester initiates the operation by sending a message to express his/her intent to request ads. The Agent who receives the message first checks his/her availability by looking at different parameters such as his/her free buffer space and then responds either with a negative or positive reply that is also accompanied with his/her TTA_{Agent} and the total number of requests that he/she is willing to take.

Upon receiving the TTA_{Agent}, the Requester can use it in combination with the $EA_{(Requester,Agent)}$ ($EA_{(R,A)}$ for short) that he/she already possesses, to calculate an DT_{Agent}. The DT_{Agent} can be calculated as follows: if the current time is $T_0 = 0$, the Requester can assume that his/her future encounters with the Agent are expected to occur between intervals of time that are equal to $EA_{(R,A)}$. We can therefore approximate the time of expected future encounters as $T_1 = T_0 + EA_{(R,A)}, T_2 = T_1 + EA_{(R,A)}, ..., T_n = T_{(n-1)} + EA_{(R,A)}$.

Delivery will occur upon one of these encounters and more specifically upon the encounter that will take place right after the Agent has visited the Ad-Dealer. The DT_{Agent} is therefore equal to $T_n > TTA_{Agent}$ when $T_{(n-1)} < TTA_{Agent}$. Time $T_{(n-1)}$ is when the Requester and the Agent meet for the last time before the Agent collects ads from the Ad-Dealer; therefore, it is smaller than TTA_{Agent}, which is the time that the collection is expected to take place. Time T_n is the time when the Requester and the Agent first meet after the Agent has made the collection; therefore T_n is greater than TTA_{Agent}. Note that if $EA_{(R,A)}$ is greater than TTA_{Agent}, then $T_{(n-1)} = T_0$ and delivery is expected to occur on the following meeting ($DT_{Agent} = T_1$).

In the scenario in Figure 3, an encounter between the Requester and the Agent takes place at $T_0 = 0$. Based on the Encounter Average $EA_{(R,A)}$ which is 10 h, future encounters can be predicted to happen between intervals of approximately 10 h ($T_1 = 10, T_2 = 20, T_3 = 30$). Delivery is expected to occur on the meeting that follows the Agent's visit to the Ad-Dealer that is anticipated to take place 24 h later, as per TTA_{Agent}. Consequently, DT_{Agent} is equal to the time of the first meeting that will take place after a period that is greater than TTA_{Agent}. The $TTA_{Agent} = 24$, therefore, $DT_{Agent} = T_3 = 30$ ($T_3 > TTA_{Agent}$) as the previous encounter $T_2 = 20$ is less than TTA_{Agent}.

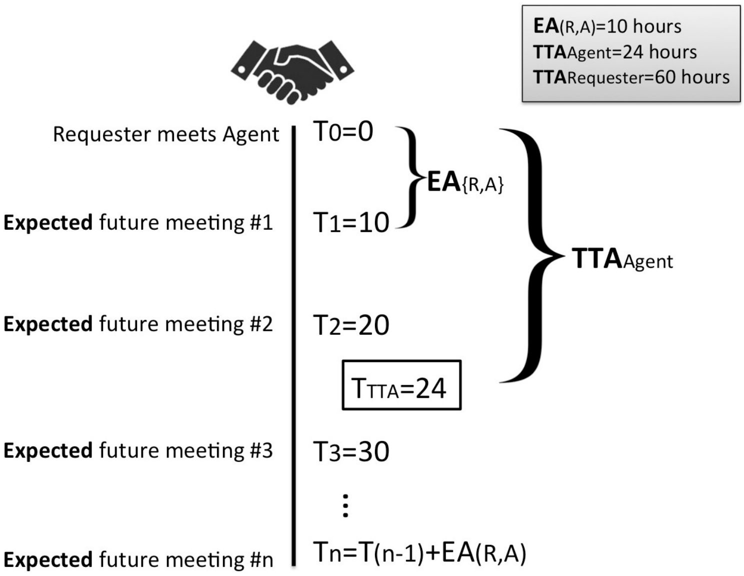


Figure 3. Calculation of DT_{Agent} (Agent's Delivery Time).

The Requester is now able to decide whether or not to use the services of that particular Agent by comparing the Agent's DT_{Agent} with the average time that it takes him/her to collect adverts, i.e., his/her own $TTA_{Requester}$. In this particular scenario, the Requester visits the Ad-Dealer between average time intervals $TTA_{Requester} = 60$ h, which is significantly more than the Agent's $DT_{Agent} = 30$.

Based on this observation, the Requester can deduce that it is within his/her best interest to send his/her requests through that Agent.

5.3.2. UIP Comparison of Requester and Agent

If the Requester decides to use the services of the Agent, he/she then may select a number of specific interests that he/she will request adverts for. In an effort to minimize bandwidth and memory overhead, the Requester attempts to give priority to adverts that the Agent is also interested in. This way, a single advert can be downloaded and viewed by both users.

Interest commonalities could be easily identified by openly comparing the UIPs of the two users, but this would also mean that the users would need to expose their advertising interests to each other. To maintain privacy, the system implements a probability-based UIP comparison mechanism. This mechanism enables the Requester to select some of his/her interests which have a high probability of also being shared with the Agent, but still prevents both of them from learning each other's exact interests.

The UIP, which includes both marked and unmarked interests, is first split into smaller groups of entries, and each of the groups is given a unique name (A, B, C, etc.). The number and size of the groups is determined by the system administrator who is the Broker, and it can vary depending on the total size of the supported UIP, which determines the number of different interests categories that are supported by the system. The size of the groups is a compromise between accuracy and privacy as smaller groups will produce more accurate results while larger groups make it more difficult for a curious Agent to determine the Requester's exact interests. The Agent then composes a list with the group names in descending order based on the number of marked interests in each group and sends it to the Requester. For example, the interest profile UIP that is shown in Figure 4 consists of 50 marked or unmarked interests that are divided into five groups (A, B, C, D and E) with a total of 10 entries in each group. If the Agent were to have the most marked interests in Group C followed by D, then E, A and, finally, B, then his/her composed list L1 would be as L1 = [C,D,E,A,B].

Upon receiving L1, the Requester follows the same procedure with his/her own $UIP_{Requester}$ in order to produce L2 = [B,A,C,D,E]. The Requester can now select a group by simultaneously going through the items in both lists from the top ranked to the bottom ranked until he/she finds a group in L2 that has the same or higher ranking in L1. More specifically, the Requester will begin by comparing the top ranked groups in each list (Group C in L1 and Group B in L2). If they do not match, he/she will then compare the top two items of each list and so forth.

In our particular example, the first comparison will be between L1′= [C] and L2′ = [B]. As a match is not found, the second comparison will take place between L1′ = [C,D] and L2′ = [B,A] that also does not produce a match. The third comparison will be between L1′ = [C,D,E] and L2′ = [B,A,C] to which the system will identify the leading group to be C, as it has the third highest ranking in L2 and the top ranking in L1. Note that if we had L1′ = [A,B] and L2′ = [B,A], then the leading group would have been A, as we always select the common item that has the highest ranking in the Agent's list L1.

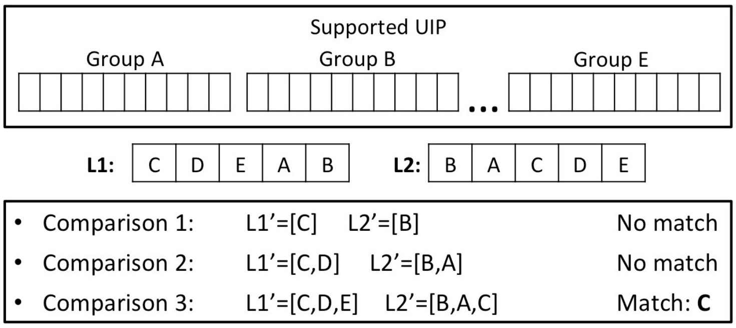

Supported UIP
Group A
Group B
Group E
...
L1: C D E A B
L2: B A C D E
• Comparison 1: L1'=[C] L2'=[B] No match
• Comparison 2: L1'=[C,D] L2'=[B,A] No match
• Comparison 3: L1'=[C,D,E] L2'=[B,A,C] Match: **C**

Figure 4. Example of algorithm for User's Interest Profile (UIP) comparison.

Once the leading group has been identified, the Requester picks N marked interests from that group. N is sent to the Requester by the Agent and is the number of requests that the Agent is willing to accept. Should N be greater than the total number of marked interests that the Requester has in the leading group (Group C), then the Requester would continue the algorithm until he/she finds a second group and supplements the remaining interests from there.

5.3.3. Transmission of ARMs

After the Requester has determined which interests to request adverts for, he/she retrieves the corresponding ARMs. Recall that ARMs and the matching keys K_i^{user} have already been composed and stored in memory. The ARMs are labelled with a unique identifier that is called the ARM-ID, and they are ultimately sent to the Agent. Finally, the Requester places his/her copy of the ARMs and K_i^{user} alongside the ARM-IDs, the User-ID of the Agent and a timestamp on a separate queue of pending requests and waits for the requester to return with the requested adverts.

5.4. User Meets Ad-Dealer

Ad-Dealers can dispatch adverts to users within their proximity over anonymous connections through the use of specialized networking equipment, which is similar to that being used in [10,14], which we assume is secure.

5.4.1. Ad-Dealer Authentication and Logging

Before logging their encounter with an Ad-Dealer, users must authenticate the Ad-Dealer. Installed within the user's client is an additional asymmetric verification (public) key K_{auth}, which is the same for all clients, and it is used just for authenticating Ad-Dealers. The matching signing (private) key K_{sign} is given by the Broker to the Ad-Dealers. As shown in Figure 5, users authenticate an Ad-Dealer by sending a random nonce R, which the Ad-dealer must sign with K_{sign}. Since K_{sign} is kept secret among the Ad-Dealers, authentication can be achieved by attempting to verify the signature on the signed R with the use of the K_{auth}.

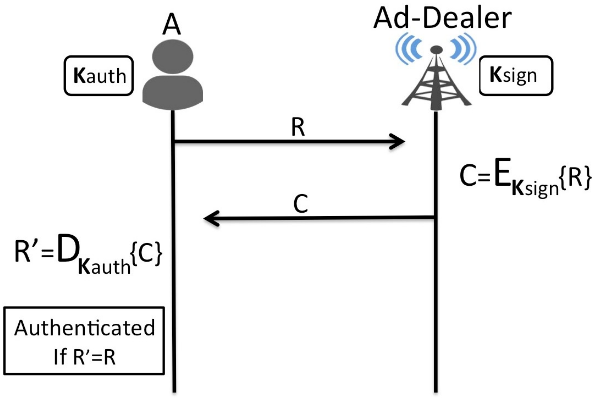


Figure 5. AD-Dealer authentication protocol.

5.4.2. Construction of Bundles and ADMs

After authentication has been successful, the users may log their encounter with the Ad-Dealer and proceed to forward to him/her both the ARMs they constructed themselves, as well as those they received from their Requesters. The AD-Dealer then decrypts the ARMs with DK_{System} and fetches adverts for the requested interests. Depending on availability, each particular interests may be matched to one or multiple ads.

Since the Requester's ARMs are selected after a profile comparison with the Agent, it is to be expected that both users may have multiple requests for the same ads. A significant amount of memory and bandwidth can therefore be conserved by distributing these adverts as a single copy that is addressed to both users. However, this would expose each other's interests and essentially defeat the purpose of performing a private profile comparison in the first place.

Our system bypasses this problem by deploying a novel protocol that enables multiple users to share access to the same ads while preventing them from knowing which ads are being shared. Specifically, the Agent and the Requester are able to view the same encrypted ads, but neither of them can learn what ads are being viewed by the other.

The Ad-Dealer initially encrypts each of the requested adverts with a different asymmetric encryption key EK_i^{AD} (Encryption Key of advert *i*) of his/her own choosing. The encrypted ads are then stamped with a unique AD-ID number, and they are organized into a sequence of groups that are called Bundles.

The Bundles are composed in such a way that shared ads, i.e., ads that are addressed to multiple users, are mixed with ads that are addressed to the respective users individually. In our particular case, where we only have two users, the Agent and the Requester, a Bundle, which contains a shared ad $AD_{(Agent,Requester)}$ ($AD_{(A,R)}$ for short), also contains some ads that are addressed only to the Agent ($AD^1_{(A)}$, $AD^2_{(A)}$), as well as some ads that are addressed only to the Requester ($AD^1_{(R)}$,$AD^2_{(R)}$). Under no circumstances should we have a Bundle that contains only shared ads or a mix of shared ads and ads that are addressed to one of the users, but no ads that are addressed to the other user. The reasons for this become evident later on when we evaluate the privacy of the system.

Once the Bundles have been composed, the Ad-Dealer creates Advert Delivery Messages (ADMs) for every one of the ARMs that he/she received. The ADMs are intended as responses to the ARMs and contain three pieces of information: (1) the AD-IDs of the adverts that relate to interest of a particular ARM, (2) the cryptographic keys DK_i^{AD} (Decryption Key of advert *i*) that can be used to decrypt them (the ads have been encrypted with the corresponding EK_i^{AD}) and (3) the sequence number of the Bundles in which the adverts are enclosed (note that multiple ads for a single interest may not necessarily be sent in the same Bundle).

Recall that every ARM is labelled with a unique ARM-ID, and it contains a user-generated encryption key K_i^{user}. Each respective ADM is therefore encrypted with the corresponding K_i^{user}, and the resulting ciphertext is labelled with the matching ARM-ID before being sent to the Agent along with all the Bundles of encrypted adverts.

$$ADM_{ARM-ID} = E_{K_i^{user}}\left[AD - ID, DK_i^{AD}, BUNDLE_{(n)}\right] \tag{3}$$

5.5. User Obtains Ads

5.5.1. Advert Collection from Ad-Dealer

Upon obtaining the ADMs from the Ad-Dealer, the user can use K_i^{Agent} to decrypt the ones that are addressed to him/her, i.e., the ADMs that have the same labels as his/her ARMs. The user can now use the information that was contained in each ADM to locate his/her adverts within the appropriate Bundles and then decrypt them with the DK_i^{AD} that were also contained in the ADM. If the user operates as an Agent, he/she stores the remaining encrypted ADMs that are not addressed to him until he/she gets the chance to deliver them. These are the ADMs that have the same labels as ARMs that he/she forwarded on behalf of a Requester.

5.5.2. Advert Reception through an Agent

When the Agent encounters the Requester after a collection has been made and after the two users have authenticated each other and logged the meeting, the Requester sends an inquiry about the state of the order to which the Agent can respond by transferring the remaining encrypted ADMs. The Requester decrypts the ADMs and obtains the AD-IDs of his/her requested adverts, the sequence numbers of the Bundles they are located in and the DK_i^{AD} keys that can be used to decrypt them. The

Requester can now ask for the specific Bundles, locate his/her ads and then decrypt them. Once the process has been deemed successful, the Requester stores the adverts for future viewing and removes the corresponding ARMs and K_i^{user} from his/her queue of pending requests, where they were placed when the request was made. In the event that an Agent fails to deliver after a certain period of time has passed that is determined by the Requester, then the Requester may reinstate the ARMs from his/her queue of pending requests and proceed to forward them to a different Agent.

6. Evaluation

We evaluate our system by performing a direct security comparison to the relevant work. In Section 3, we examine the currently available systems and identify security risks and shortcomings. The identified shortcomings are taken into account in Section 4.3 where we define a set of operational requirements that will serve as evaluation criteria for our system. For our evaluation, we first determine certain threat scenarios under which the current models would fail to meet our requirements and then use the same scenarios to examine the performance of our own system.

6.1. User Privacy against Other Parties (Broker and Ad-Dealers)

Similarly to the architectures that are described in [11,18], which we assume as secure, our model enables users to establish anonymous connections with a broadcasting station and download content without exposing any identifying information. From that perspective, our system offers a very similar level of protection, as the Ad-Dealer and Broker have no means of obtaining any information that may reveal the user's physical identity, such as username, email and home IP address.

The main difference of our system is that users are also downloading content on behalf of others within their social network. The content is encrypted, and therefore, our system requires the transmission of cryptographic keys, which provides to the Ad-Dealer additional information about the users. The use of cryptography may therefore have implications on privacy, which we will now proceed to examine.

Recall that every ARM that is sent to the Ad-Dealer contains a user-generated key K_i^{user} that is later used by the Ad-Dealer to encrypt the corresponding ADM. Furthermore, the keys are produced by a hash chain where $K_i^{user} = h(K_{i-1}^{user})$. We assume that the Ad-Dealer has no way of obtaining the original key K_0^{user}, which was computed by the user, but knows which hash function $h()$ is used.

Based on this knowledge, the Ad-Dealer can obtain K_i^{user} out of one ARM and then replicate the hash chain in order to compute all consecutive keys K_{i+1}^{user}, K_{i+2}^{user}, etc. without knowing K_0^{user}. This means that the Ad-Dealer can identify ARMs that contain subsequent keys, and from that, he/she may deduce that they have all been created by the same user even if they are sent through multiple Agents and at different time instants. Therefore, the key can be used as an identifier to profile the users. However, this does not compromise privacy to a significant extent as the Ad-Dealer still has no means of learning the user's physical identity. This threat can be minimized if the user changes K_0^{user} very frequently, but this will require the consumption of additional processing power.

One more aspect that needs to be considered is the fact that the Ad-Dealer is also able to infer information about the social interactions of users and, more specifically, the affiliations between Requesters and Agents. For as long as the identities of the users are kept private, this also has minimal implication to privacy, but we do need to consider the possibility of this not always being the case. Should the Ad-Dealer find a way to uncover the true identities of users, then not only can he/she associate it with their advertising interests, but he/she will also learn about their social connections with each other.

The only way of achieving this is by compromising the software client that runs on the user's mobile device. Broker interference with the client would be classified as malicious behaviour; hence, it is out of scope in our case, as we assume an honest-but-curious Broker. Even if the Broker were malicious, tampering with the software would be easily detected and would have serious legal implications. The integrity of the software can be certified by an independent authority such as the developer of the platform it operates on (iOS, Android, Windows mobile).

6.2. User Privacy against Other Users

As adverts are conveyed between peers within a social network, privacy among users is the prime concern that has been mainly dismissed in previously proposed systems. Note that an interaction between two users can be divided into four phases that will now be scrutinized separately.

6.2.1. Requester and Agent Compare UIPs

During the initial phase, a potential Requester and an Agent perform a comparison of their respective interest profiles (UIPs). Recall that the UIP is first separated into groups of entries, and the Agent discloses the group names in descending order based on the number of marked interests he/she has in each group. At this point, the Requester can infer that the Agent has more interests in a high-ranking group than a low-ranked one, but he/she is still completely oblivious to what these interests are or even how many they are. The Agent can assume that the Requester's selection will most likely be from a high-ranking group. Other than this trivial knowledge, the users do not learn any more information about each other's interests, and we can therefore conclude that privacy is preserved.

6.2.2. Agent Receives ARMs from Requester

As has already been stated, ARMs are encrypted with the EK_{System} (System Encryption Key) and, therefore, cannot be read by the Agent, who has no knowledge of the corresponding decryption key DK_{System}. Recall that ARMs contain an Interest-ID, which is unique for the entire system and known to all users, but also contain a key K_i^{user}, which is generated by the user. This essentially means that any two ARMs that contain the same Interest-ID are highly unlikely to produce the same cryptograms as the K_i^{user} is also included as part of the plaintext. An encrypted ARM is therefore secure as it would be infeasible for an attacker to guess its content by comparing it to other encrypted ARMs that he/she creates himself/herself (chosen plaintext attack).

6.2.3. Agent Collects Ads

When appearing within the proximity of an Ad-Dealer, the Agent forwards all of his/her stored ARMs, including his/her own. The Ad-Dealer fetches the requested adverts, encrypts them with different keys of his/her own choosing and he/she organizes them in Bundles. For every ARM, the Ad-Dealer then creates an ADM (Advert Delivery Message), which contains information that will allow the recipient to locate his/her ads within the appropriate Bundle and decrypt them. Finally, the ADMs are encrypted with K_i^{user} and sent back to the Agent. The Agent is able to decrypt the ADMs that are responses to his/her own ARMs, since he/she knows K_i^{Agent}, but the Requester's ADMs remain private as the Agent cannot decrypt them without knowing the $K_i^{Requester}$.

6.2.4. Agent Delivers Ads

Upon meeting the Agent, the Requester first receives his/her ADMs. Then, he/she asks for the Bundles where his/her ads are stored based on the information he/she obtains from the ADMs. Finally, he/she recovers his/her ads, which can either be ads that are meant just for him/her or shared ads, i.e., ads that are addressed to multiple users.

Recall that shared ads are placed within Bundles in such a way that they are mixed with other ads which are addressed to the respective users individually.

Ads that are addressed to only one user, either the Agent of the Requester, remain completely private from the other user who cannot decrypt them without having the appropriate decryption key DK_i^{AD}, which can only be obtained from an ADM. However, the shared ads can be read by both users, and therefore, if one of the users is able to identify which of the ads within a Bundle are shared, he/she will immediately learn the interests of the other user.

Upon receiving a Bundle, the Requester has no means of identifying the shared ads, as they are mixed with the other ads, and therefore, the privacy of the Agent and the privacy of other Requesters who use the same Agent are preserved.

From the Agent's perspective, when the Requester asks for a specific Bundle, then the Agent can assume there is a possibility that some of the ads within the Bundle may be shared. Despite this, the Agent has no way of being absolutely certain of that possibility, as the Requester may be asking for the Bundle just because he/she wants to obtain his/her own ads, which are individually addressed ads that are not visible to the Agent.

If the demanded Bundle were to include no individually-addressed ads for the Requester, the entire content of the Bundle would be visible to the Agent. Then, the Agent would have been able to infer that at least one of the contained ads is shared and that would put the Requester's privacy at risk. For this reason, the Ad-Dealer composes Bundles is such a way that this is never the case. Even in the extreme scenario where the exact same ARMs were sent by both users and, therefore, all of the ads can be shared, the Ad-Dealer prevents this from happening by supplementing additional ads.

One last possibility we need to consider is that a user acts completely maliciously by fabricating his/her UIP, i.e., he/she carefully composes his/her UIP in such a way that only specific interests are marked. A malicious Requester would pose no threat for an Agent as the former does not obtain any information that would enable him/her to identify the shared adverts within a Bundle even if the Requester's UIP is fabricated. A malicious Agent, however, may threaten the privacy of a Requester by employing a fabricated UIP.

Consider a scenario where a malicious Agent engages in a profile comparison with a potential Requester by listing the groups of his/her UIP in order L = [A,B,C,D,E], and the Requester sends him/her two ARMs. At this point, the Agent may assume that it is highly likely that the ARMs are referring to interests that can be found in one of the high-ranked groups, such as A or B. The Agent then composes $ARM_{(x)}$ for a specific interest x, which can be found in one of his/her highly-ranked groups, such as Group A. Afterwards, he/she creates multiple additional ARMs for more interests that belong to low-ranking groups (e.g., Group E or D) and then sends all of them to the Ad-Dealer. If $ARM_{(x)}$ is the same as one of the two ARMs that were sent by the Requester, the Ad-Dealer will retrieve a single shared advert and address it to both users. The Ad-Dealer will also fetch additional adverts for the Requester's second ARM, as well as for the remaining ARMs of the Agent. Since both the Requester and the Agent have at least one individually-addressed advert, the Ad-Dealer can compose all of the ads into a single Bundle and send it back to the Agent. When receiving the Bundle, the Agent should be able to access all of the contained ads except for one: the advert that was requested by the Requester, but not by himself/herself. Based on the fact that the Requester has sent two ARMs, the Agent can infer that within the Bundle, there is one shared advert.

Recall that during the profile comparison operation, the Requester was manipulated into selecting interests out of high-ranking groups (Group A or B), but the Agent requested only advert x out of these groups and all his/her remaining ARM were deliberately selected from low-ranking groups. Based on this, the Agent can deduce that the shared advert concerns interest X.

This type of attack is highly impractical as it is only limited to one interest at a time and cannot work when the Requester sends a larger number of ARMs and the Ad-Dealer responds to each of them with multiple ads that are composed into different Bundles. We can therefore conclude that this attack has a very small chance of being successful, and even when it succeeds, the effects on the Requester's privacy are not very severe.

6.3. Protection from Malicious Content

The spread of fake or harmful content is a serious threat that is not addressed by previously-proposed advert distribution systems that make use of opportunistic technologies. As data are propagated through temporary connections among peers, a malicious Agent could easily replace an advert with his/her own content before forwarding it. Our system overcomes this threat through the use of cryptography on both the ARMs, as well as the ads themselves. As previously stated in Section 5, ARMs are encrypted with EK_{System}, ads are encrypted with EK_i^{AD} and the corresponding DK_i^{AD} is composed into an ADM, which is encrypted with K_i^{user}.

In order for an attacker to replace an advert within a Bundle, he/she would need to encrypt the new malicious advert with the same EK_Z^{AD}, which he/she does not have. If the attacker were to simply use a new encryption key $EK_{Attacker}$ that he/she generates himself/herself, the other users would not be able to decrypt the malicious advert since they do not have the corresponding decryption key $DK_{Attacker}$. For this attack to work, the attacker would first need to create a new ADM that contains the appropriate $DK_{Attacker}$, and then, the new ADM would need to be encrypted with K_i^{user}. Obtaining K_i^{user} however is impossible, as it is enclosed in the ARM that has been encrypted with EK_{System} and can therefore only be accessed by the Ad-Dealer who knows DK_{System}.

One last possibility that needs to be considered is that of the user being tricked into creating ARMs that are not encrypted with EK_{System}. Since EK_{System} is installed within the client software, this could only be possible if the attacker managed to compromise the user's mobile device. If the attacker were to have access to the user's client, then this attack would be trivial since the attacker could easily bypass the system by simply replacing the locally stored adverts. We can therefore conclude that for as long as DK_{System} and EK_i^{AD} are kept secret, it is infeasible for anyone other than an Ad-Dealer to send encrypted adverts to users.

6.4. Protection from Impersonation Attacks

A malicious user can launch an attack against the system by impersonating the identity of a legitimate system stakeholder, either a user or an Ad-Dealer. The four possible scenarios that may accrue from such an attack are the following.

- Victim is a user and the Attacker impersonates one of his/her Contacts (that is not an Agent or a Requester).
- Victim is a Requester and the Attacker impersonates his/her Agent
- Victim is an Agent and the Attacker impersonates his/her Requester
- Victim is an Agent and the Attacker impersonates the Ad-Dealer

In the first scenario, the attacker tricks a user U1 into registering a fake encounter with one of his/her Contacts U2. This would alter the Encounter Average $EA_{(U1,U2)}$ of the two users and in turn interfere with the correct operation of the opportunistic routing algorithm.

In the scenario where the attacker impersonates an Agent there are two cases. In the first case, the victim is a potential Requester, i.e., a user who has not yet made a request, who is manipulated into sending ARMs to the attacker. The attacker would still be unable to read the encrypted ARMs, but this would also result in the requests being lost. In the second case, the victim is already a Requester, i.e., the victim has placed a request and is waiting for ads. If this happens, the Requester may be tricked into downloading data, such as fake ADMs and Bundles, that have not been sent by a legitimate Ad-Dealer.

In the third scenario, the attacker impersonates the Requester, after the request has been made, to an Agent who is tricked into sending to the attacker the adverts (ADMs and Bundles) that are meant for the real Requester. After completing the delivery, the Agent (victim) will have no reason to keep the ADMs, which will be discarded. This will result in a denial of service for the real Requester, who will not be able to obtain his/her adverts when the actual meeting between him and the Agent takes place.

These first three scenarios are prevented with the use of password-based mutual authentication. The authentication protocol uses a standard challenge-response handshake that is assumed to be secure as long as the passwords are not leaked. The original exchange of passwords happens through an out-of-band channel (e.g., manual input, QR code, text) which is very difficult for an attacker to eavesdrop on. Although the passwords are generated automatically by a random function within the client, we assume that they cannot be guessed by an attacker, as the function may include several sources of randomness from the user's device, such as memory usage, time or signal strength.

For our final scenario, the attacker impersonates the identity of an Ad-Dealer. Nearby Agents would be deceived into sending to the attacker their ARMs. The ARMs are encrypted and the attacker would not be able to use them, nor would he/she be able to send back any malicious content. However, the ARMs could potentially be lost, which would result in failure to deliver on behalf of the Agent. Agents avoid this attack by authenticating the Ad-Dealer with the use of digital signatures. The Agent sends a random nonce that the Ad-Dealer is asked to encrypt with the K_{sign} and can be verified with the K_{auth}. We need to note that this creates a threat for the Ad-Dealer being tricked into signing something else other than a user's nonce. However, this would serve no purpose since the Ad-Dealer's signature is used only for this operation.

6.5. Resource Conservation

Wasting of resources is a typical problem of opportunistic networking schemes as multiple copies of the same data exist across the network. Consider a simple example where an Agent receives three different encrypted requests from Requesters R1, R2 and R3 and all of them are for the same advert. When visiting the Ad-Dealer, the Agent would need to collect three copies of the same advert, each copy encrypted with a different key for each Requester. This increases overhead on the Agent side who needs to download and store excess data.

Our system reduces this overhead by taking advantage of the multicast nature of advert delivery as the same advert is relevant for multiple consumers. Instead of circulating multiple copies of the same advert to different users, we are able to spread a single copy that is addressed to multiple recipients. What is innovative about our design is that we achieve this without compromising privacy, as we have already demonstrated in previous sections.

Interest commonalities are expected to emerge naturally among social groups who share the same demographics (age, religion, nationality, etc.) [19]. To further enhance this effect, our system attempts to identify specific shared interests by comparing the user's UIPs. Although it may be counter-intuitive, our aim here is not to achieve complete accuracy. A completely accurate discovery of common interests could compromise privacy among Agents and Requesters. The probabilistic approach that we follow preserves privacy, but at the same time, increases the chances of a randomly selected interest being shared by two users.

This is better illustrated in a simple example. Consider a scenario where a potential Requester wishes to place a request to an Agent. The Agent's interest profile UIP consists of 30 entries out of a total of 100 interests that are supported by the system. This means that if the requester were to perform a random selection, he/she would have a 30% chance of selecting one of the Agent's interests. When we apply our algorithm, the profile is segmented into five groups of twenty interests. The groups are then ranked into descending order which the highest ranked containing 15 interests, the second highest 10, the third five and the two remaining groups containing zero. This enables the Requester to effectively focus his/her priorities on the higher ranked groups and ignore the rest. This would give him a 75% chance of success for the top ranked group, 50% for the second highest ranked group and 25% for the third.

Naturally, these numbers will vary depending on how sparsely the Agent's interests are spread across the UIP. The total number and size of the groups can be altered depending on the number of supported interests and the desired level of accuracy. In addition to this, the groups can potentially be composed based on logical criteria, e.g., each group containing interests that are relevant to specific types of consumers. For example, a group may include interests that are commonly associated with male consumers while another group will include interests that are common among females. This is very similar to the profiling method that is currently used by advertising companies and will result in greater accumulation of marked entries within specific groups. However, in order to maintain privacy, the groups need to be created based on criteria that do not expose any information that the Requester and Agent do not already know about each other. This may include demographic information that is very generic such as gender, age and nationality.

7. Conclusions and Future Work

In this paper, we present an advert distribution system, which combines ad-broadcasting (anonymous download of ads from broadcasting stations) with opportunistic networks. We first classify and scrutinize relevant systems in terms of privacy, security, targeting effectiveness and practicality and then recognize their shortcomings. For our evaluation, we use the identified shortcomings of relevant systems in order to examine the effectiveness of our own system under different threat scenarios. Our evaluation shows that our system achieves a greater level of privacy by eliminating the need for trust among users, which are nodes of the opportunistic network, and at the same time, reduces memory and bandwidth overhead by allowing multiple users to share access to the same ads without compromising their privacy. Counter to previous models, our system considers the possibility of malicious activity and effectively prevents the spread of fake ads from a rogue node, as well as impersonation attacks.

We are currently in the development stage of a working prototype, which will enable us to experimentally assess the delivery performance and the resource efficiency of our model, including memory footprint and impact on battery. Additionally, we intend to incorporate a secure click-report mechanism that will prevent fraud against the Advertisers, eliminate the need for a Broker and, at the same time, work as a reward system, which will provide incentives for the participation of Agents and Ad-Dealers.

Author Contributions: Stylianos Mamais, security design and manuscript preparation. George Theodorakopoulos, initial idea, suggestions on cryptography.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. eMarketer. Mobile to Account for More than Half of Digital Ad Spending in 2015. Available online: https://www.emarketer.com/Article/Mobile-Account-More-than-Half-of-Digital-Ad-Spending-2015/1012930 (accessed on 18 January 2017).
2. The Guardian. UK Mobile Ad Spend 'To Overtake Print and TV'. Available online: https://www.theguardian.com/media/2015/sep/30/mobile-advertising-spend-print-tv-emarketer (accessed on 18 January 2017).
3. Yan, J.; Liu, N.; Wang, G.; Zhang, W.; Jiang, Y.; Chen, Z. How much can behavioural targeting help online advertising? In Proceedings of the 18th International Conference on World Wide Web, Madrid, Spain, 20–24 April 2009; pp. 261–270.
4. Purcell K.; Brenner J.; Rainie L. Pew Research Center: Search Engine Use 2012. Available online: http://www.pewinternet.org/2012/03/09/search-engine-use-2012/ (accessed on 18 January 2017).
5. Federal Trade Commission (FTC). FTC Staff Report: Self-Regulatory Principles For Online Behavioural Advertising: Tracking, Targeting, and Technology. Available online: https://www.ftc.gov/sites/default/files/documents/reports/federal-trade-commission-staff-report-self-regulatory-principles-online-behavioural-advertising/p085400behavadreport.pdf (accessed on 18 January 2017).
6. Sun, Y.; Ji, G. Privacy preserving in personalized mobile marketing. In *Active Media Technology: 6th International Conference, AMT 2010, Toronto, Canada, August 28–30, 2010 Proceedings*; Springer: New York, NY, USA, 2010; pp. 538–545.
7. Backes, M.; Kate, A.; Maffei, M.; Pecina, K. Obliviad: Provably secure and practical online behavioural advertising. In Proceedings of the 2012 IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 21–23 May 2012; pp. 257–271.
8. Chor, B.; Goldreich, O.; Kushilevitz, E.; Sudan, M. Private information retrieval. In Proceedings of the IEEE 36th Annual Symposium on Foundations of Computer Science, Milwaukee, WI, USA, 23–25 October 1995; pp. 41–50.

9. Toubiana, V.; Narayanan, A.; Boneh, D.; Nissenbaum, H.; Barocas, S. Adnostic: Privacy preserving targeted advertising. In Proceedings of the Network and Distributed System Symposium, San Diego, CA, USA, 28 February–3 March 2010.
10. Kodialam, M.; Lakshman, T.; Mukherjee, S. Effective ad targeting with concealed profiles. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), Orlando, FL USA, 25–30 March 2012; pp. 2237–2245.
11. Guha, S.; Cheng, B.; Francis, P. Privad: Practical privacy in online advertising. In Proceedings of the 8th USENIX conference on Networked Systems Design and Implementation, Boston, MA, USA, 30 March–1 April 2011; pp. 169–182.
12. Carrara, L.; Orsi, G. *A New Perspective in Pervasive Advertising*; Technical Report; Department of Computer Science, University of Oxford: Oxford, UK, 2011.
13. Carrara, L.; Orsi, G.; Tanca, L. Semantic pervasive advertising. In Proceedings of the 7th International Conference on Web Reasoning and Rule Systems, Mannheim, Germany, 27–29 July 2013; pp. 216–222.
14. Straub, T.; Heinemann, A. An anonymous bonus point system for mobile commerce based on word-of-mouth recommendation. In Proceedings of the 2004 ACM Symposium on Applied Computing, Nicosia, Cyprus, 14–17 March 2004; pp. 766–773.
15. Ratsimor, O.; Finin, T.; Joshi, A.; Yesha, Y. eNcentive: A framework for intelligent marketing in mobile peer-to-peer environments. In *The 5th international conference on Electronic Commerce (ICEC 2003)*; ACM: New York, NY, USA, 2003; pp. 87–94.
16. Ntalkos, L.; Kambourakis, G.; Damopoulos, D. Let's Meet! A participatory-based discovery and rendezvous mobile marketing framework. *Telemat. Inform.* **2015**, *32*, 539–563.
17. Lindgren, A.; Doria, A.; Schelen, O. Probabilistic routing in intermittently connected networks. In *Service Assurance with Partial and Intermittent Resources*; Springer: New York, NY, USA, 2004; pp. 239–254.
18. Haddadi, H.; Hui, P.; Brown, I. MobiAd: Private and scalable mobile advertising. In Proceedings of the Fifth ACM international Workshop on Mobility in the Evolving Internet Architecture, Chicago, IL, USA, 20–24 September 2010; pp. 33–38.
19. Lazer, W. *Handbook of Demographics for Marketing & Advertising: New Trends in The American Marketplace*; Lexington Books: New York, NY, USA, 1994.