future internet

Article

Behavioural Verification: Preventing Report Fraud in Decentralized Advert Distribution Systems

Stylianos S. Mamais *,† and George Theodorakopoulos †

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

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

† Current address: School of Computer Science and Informatics, Cardiff University, 5 The Parade, Roath, Cardiff CF24 3AA, UK.

Received:18 September 2017; Accepted: 16 November 2017; Published: 20 November 2017

Abstract: Service commissions, which are claimed by Ad-Networks and Publishers, are susceptible to forgery as non-human operators are able to artificially create fictitious traffic on digital platforms for the purpose of committing financial fraud. This places a significant strain on Advertisers who have no effective means of differentiating fabricated Ad-Reports from those which correspond to real consumer activity. To address this problem, we contribute an advert reporting system which utilizes opportunistic networking and a blockchain-inspired construction in order to identify authentic Ad-Reports by determining whether they were composed by honest or dishonest users. 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 purchasing goods online, moving through space and interacting with other users. We argue that it is hard for dishonest users to fake honest behaviour and we exploit the behavioural patterns of users in order to classify Ad-Reports as real or fabricated. By determining the honesty of the user who submitted a particular report, our system offers a more secure reward-claiming model which protects against fraud while still preserving the user's anonymity.

Keywords: advertising; click report; fraud prevention; mobile networks; privacy-preserving advertising

1. Introduction

Digital advertising is a form of marketing which capitalizes on the popularity of digital media Publishers such as websites and software applications in order to present consumers with promotional material from retailers who are referred as Advertisers. Ad-Networks such as Google AdSense and Yahoo! AdNet, operate as a middleman between Advertisers and Publishers. Their purpose is to generate revenue by presenting users who browse a Publisher's platform with relevant adverts of the Advertiser's products and services. In more detail, Ad-Networks are granted full control of certain areas within the Publisher's interface for the purpose of featuring adverts. When a user accesses a Publisher's digital domain (whether website or software application), a request is sent to the Ad-Network who selects the most prominent advert from one of the Advertisers and displays it along side the Publisher's content where it may be viewed by the user. The Ad-Network then claims a monetary reward by filing a report to the corresponding Advertiser and afterwards awards a commission to the Publisher for providing his platform.

The ever increasing expansion of the marketing industry makes digital advertising highly profitable for the Ad-Networks and it provides for a tenable source of profit for Publishers that is also beneficial for users who can get services at a lower price. It should therefore come as no surprise that according to eMarketer, the total spending on digital advertising in the US is estimated to exceed

$77 billion for the year 2017 with most of that money being spent specifically on adverts that target mobile users [1].

Despite being highly beneficial for all stakeholders, the Ad-Network advertising model is far from ideal as it is susceptible to fraud. According to a report released by the Investigative Advertising Bureau (IAB) in December 2015, the cost of digital advertising fraud is estimated to be over $8.2 billion with 56% of that (up to $4.6 billion) being attributed to non-human traffic [2]. Ad-Networks attempt to limit the problem through the enforcement of policy-based filtering mechanisms (which detect suspicious traffic), but this approach has not been entirely effective [3]. The Advertisers are the victims of the fraud and although they still benefit from digital ads, they are also forced to tolerate the contemporary state of affairs as they have no other choice but to trust the Ad-Networks.

This research introduces an advert reporting system which utilizes opportunistic networking, blockchain-inspired architecture and client-side processing to entirely substitute Ad-Networks with a community of independent advert-distributors called **Ad-Dealers**. The Ad-Dealers have a physical presence in publicly accessible areas and serve as anonymous communication gateways to Advertisers. Users who appear within proximity of an Ad-Dealer are able to establish a secure connection with the Advertisers in order to transfer information. Opportunistic networking is exploited to further extend the reach of the Ad-Dealers over a social network of mobile users who voluntarily ferry data for each other, while still maintaining privacy through the use of encryption. This enables the system to propagate adverts from the Ad-Dealers to targeted users but also **allows the users to anonymously submit Ad-Reports which they compose themselves**.

Although the majority of users have no immediate benefit from submitting fraudulent Ad-Reports, this does not ensure that a filled report corresponds to real user activity. As users need to remain anonymous, identifying dishonest (malicious) users through traditional methods such as digital signatures is problematic. To address this limitation, our contribution is a mechanism which enables the verification of reports that were submitted by honest users without compromising their identity.

What constitutes a user's honesty for our system is the manner upon 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. Social activities such as making online purchases, moving through space and interacting with other users are exploited by our system to verify that a report was submitted by a honest user. More specifically, when users perform a particular social task (e.g., visit a designated location or meet other users) they obtain a **Token** which works as proof of the user performing the action at a specific time. To better comprehend this concept, think of a game of scavenger-hunt where players prove to have visited a location by recovering some type of artifact from the area. In a very similar fashion, users of our system collect a series of Tokens and submit them along with their Ad-Reports in the form of a **blockchain-inspired construction** which we term as the Ad-Report Chain (*ARC*). Tokens work like time-stamps which allow Advertisers to calculate the rate at which the submitted reports were composed and identify suspicious activity (dishonest users who create large volumes of reports over short time). This results in the ability to verify that an Ad-Report claim is the outcome of real user activity without need for knowing the identity of the particular user.

Our contributions are as follows:

- To the best of our knowledge, our system is the first to exploit user behavioural patterns (Behavioural Verification) for the purpose of exposing advert fraud.
- Behavioural Verification does not need user identities and therefore preserves privacy.
- A major component of detecting fraudulent behaviour is the rate at which the use creates reports. We simplify the process of calculating this rate compared to the literature. Traditional methods compute this rate by filtering to identify reports that originate from the same source while the approach that we use logs Ad-Reports in a chronological order directly at the user's device and submits them as a single message (*ARC*).

The rest of this article is organized as follows. In Section 1.1 we present the scope of our research and in Section 2 we offer insight on some of the related work in the field. In Section 3 we provide preliminary knowledge which is required in order to completely understand the functionality of our system. In Section 4 we provide a high level overview of our model by presenting the system's architecture and formalize the addressed problem. In Section 5 we provide a detailed analysis of our system and offer insight into our approach for solving the addressed problem. Finally, we evaluate our approach in Section 6 and state our conclusion in Section 7.

1.1. Research Scope

The scope of this research is to produce a secure advert reporting mechanism which (1) allows for Ad-Dealers and Publishers to claim service fees from Advertisers, (2) protects all impacted stakeholders (Advertisers, Publishers, Ad-Dealers) from financial fraud by means of preventing the forgery of Ad-Reports and lastly (3) preserves the privacy of the users against all other parties.

2. Related Work

Online fraud is a problem that takes many different forms as expressed in [4]. In our work, we focus on the specific area of advertising fraud which has been continuously expanding over the past years. 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.

The Collaborative Click Fraud Detection and Prevention (CCFDP) [5] model, 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 sever and client side and individually return probabilistic estimates of a click's legitimacy which are combined to produce an overall score. The results of the collaboration 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 click more accurately and successfully block them.

Rather that filtering our fraudulent clicks, Juels et al. [6] 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 which 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 way that the users' identity is substantially protected against a curious adversary and also offers protection against coupon-replay attacks.

Haddadi [7] argues that click fraud is progressively becoming harder to detect through traditional threshold techniques (identifying multiple reports from the same IP) 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 from 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 may be an indicator of suspicious activity. Although we find this idea to be very creative, we need to remark upon the fact that Bluff-ads are unlike to be adopted as they take valuable space which can be used for real (profitable) ads.

Instead of operating on the server side, FCFraud [8] runs locally on the devices of individual users as a means of preventing them from being part of a BotNet. A BotNet is a group of infected devices which is 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 fake clicks and can easily be implemented in user devices but is only suitable for Advertisers who adopt the PPC (Pay-Per-Click) model.

Faou et al. [9] 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 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.

Security within mobile communications has received considerable attention in the context of both cellular [10–13] and opportunistic networks [14–16].

3. Preliminaries

The **Advert Distribution System (ADS)** [17] that we offered in our previous work, combines anonymous download technologies and opportunistic networking in order to deliver adverts to users without exposing their consumer interests. The system that we present in this paper expands the functionality of ADS by enabling users to also submit reports about the adverts that they have viewed.

As both systems share much of the same architectural elements and stakeholders, in this section we provide a fundamental overview of ADS which will be required in order to better comprehend the extended advert reporting system that will be described later on. As depicted in Figure 1, ADS establishes a communication link between **Advertisers**, who are retailers that wish to promote their products and **Users** who represent consumers. The **Ad-Dealers** are local broadcasting stations who serve as communication gateways between users and Advertisers in order to distribute adverts upon request. The role of the Ad-Dealer may be cast to regional entities such as shopping malls, WiFi hotspots and local businesses.

Users run a software client on their smartphone devices that allows them to connect anonymously to Ad-Dealers within their proximity through the use of specialized networking equipment. This allows the mobile clients to freely request and consequently download adverts which are then displayed by **Publishers** who are digital advertising platforms such as websites and mobile apps.

Furthermore, users can also communicate with the Ad-Dealers indirectly through opportunistic networking by forwarding Ad-Requests to an Agent over Bluetooth of WiFi Direct. As shown in the diagram, the Agent physically ferries the Ad-Requests to an Ad-Dealer, collects the requested adverts and conveys them back to the user. In practice, **Agents** are also mobile users who have the additional role of propagating Ad-Requests and adverts on behalf of other users within their social community. This setup benefits members of the community who do not visit Ad-Dealers on a regular basis but can obtain their adverts through other users (Agents) within their social circles.

User privacy against the Ad-Dealers is ensured by the use of anonymous connections (no use of IP address or MAC) and the fact that Agents operate as a form of partially trusted proxy between the users and Ad-Dealers. Ad-Requests are encrypted with a public key which is incorporated into the users' client software and can only be deciphered by the Ad-Dealers who share knowledge of a secret decryption key. The adverts are also encrypted by the Ad-Dealers with a key which is provided to them by the users themselves within the equivalent Ad-Requests. This preserves the privacy of users against Agents who have no access to the content of Ad-Requests nor the adverts and also guarantees security against a rogue Agent who might inject malicious adverts into the network. Security throughout the entire system is also enforced by strong authentication mechanisms as to prevent malicious users from impersonating Ad-Dealers, Agents and other users (In our previous work, we provide a detailed description of how the opportunistic network is established and the mechanisms which are employed to ensure security and anonymity. For this reason, this paper assumes existence of a secure opportunistic link and focuses exclusively on the submission of Ad-Reports).

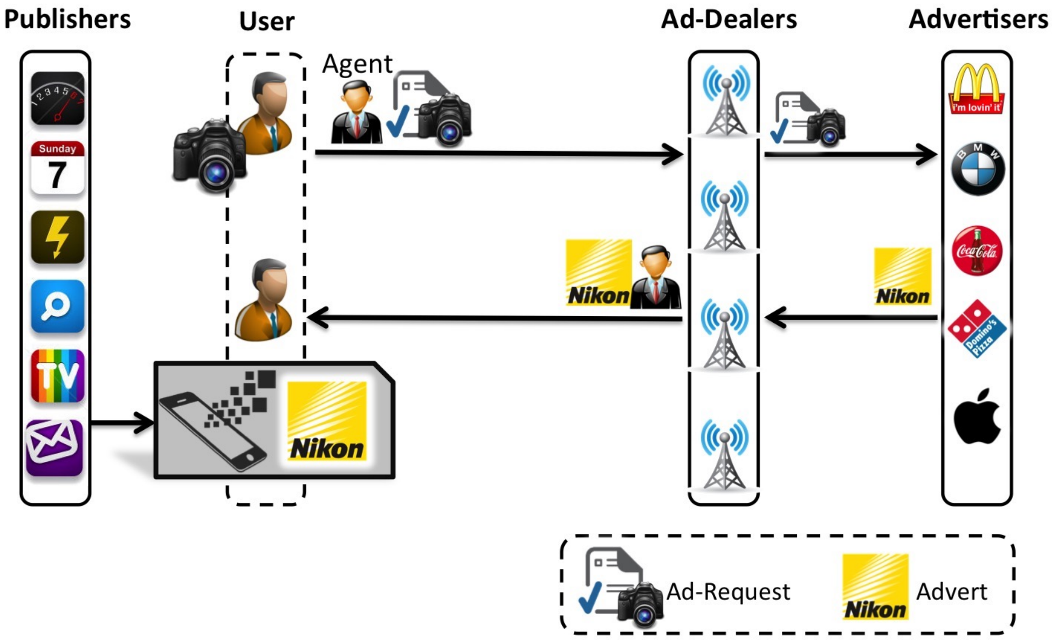

Figure 1. Decentralized advert distribution system over opportunistic network.

4. Model Overview

In this section, we provide a high level overview of our model. We begin by first featuring the stakeholders of our system and detailing their functionality in Section 4.1. We then proceed to define our adversary model in Section 4.2 by contemplating the relationships between the stakeholders in terms of trust and potential threat which enables us to formalize the problem in Section 4.3. Finally, we establish a list of evaluation criteria for our design in Section 4.4.

4.1. System Architecture

As previously mentioned, the advert reporting system that we submit with this work is an extension of our Advert Distribution System (ADS) that we briefly explained in Section 3 and therefore shares much of the same components and architecture. Figure 2 illustrates the architecture of the system and provides a high-level overview of the system's operation which can be divided into three stages with each of them performed under a different sub-protocol later examined in Section 5.3.

The Users, Publishers, Agents, Ad-Dealers and Advertisers represent the same stakeholders as in our earlier ADS system but at the same time have the added functionality of managing Ad-Reports with the help of the Broker. The Broker assumes the role of a trusted representative of the Advertisers to the Ad-Dealers and Publishers. As the Advertisers are too numerous to operate independently while still remaining coordinated, they employ the services of the Broker whose job is to collect and verify the Ad-Reports that are dispatched by the Ad-Dealers.

The user initiates the first stage that is marked in Figure 2 as **(1) Report Form or RF Collection** by composing an RF-Request the purpose of which is to inform the Broker of the user's intention to submit Ad-Reports. The RF-Request is then sent to an Agent who operates as an intermediate node of the opportunistic network. The job of the Agent is to physically transfer the RF-Request to one of the Ad-Dealers, who then forwards to the Broker. Upon receiving the RF-Request, the Broker issues a Request Form (RF) which is sent in the opposite direction: back to the Ad-Dealer, who forwards it to the Agent, so that it may be conveyed back to the requesting user. The RF contains necessary information which the user needs in order to compose and submit his/her Ad-Reports later on.

The second stage, which is marked as **(2) AD Collection**, is essentially the same advert delivery operation which we described in Section 3. Through the use of an Agent, the user sends an Ad-Request message to the Ad-Dealers in order to demand specific adverts. The Ad-Dealer who receives the Ad-Request may then obtain the requested adverts from the appropriate Advertisers and send them back to the user via the same Agent. Note that this operation can take place multiple times via different Agents and Ad-Dealers. Furthermore, it may also take place in combination with the first operation (RF Collection) as one Agent can transfer both the adverts and the RF at the same time.

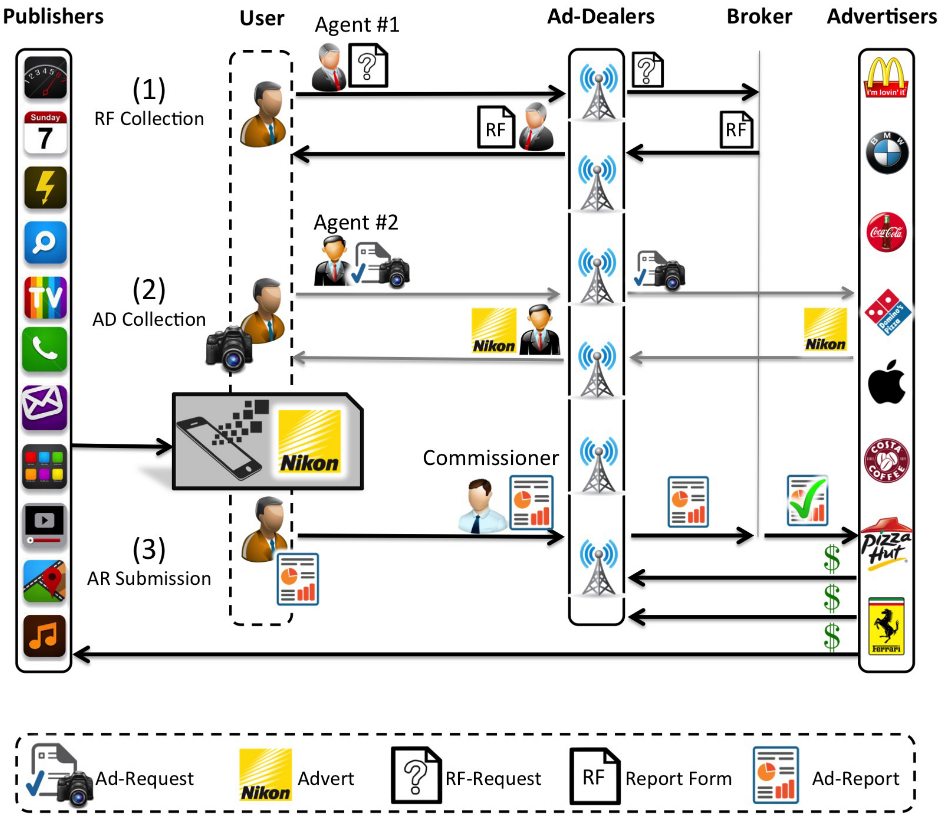

Figure 2. Decentralized advert reporting system over opportunistic network.

The final stage, which is noted as **(3) Ad Report or AR Submission**, takes place after the user has viewed the adverts through a Publisher. To notify the Broker that the user has viewed the adverts, the user composes Ad-Reports and forwards them to a Commissioner with the intent to be delivered to an Ad-Dealer. 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 eventually reward the Publishers who featured the adverts as well as the **two** involved Ad-Dealers (the first that delivered the adverts to the user and the second that submitted the Ad-Reports).

It is evident that the Commissioner serves a very similar function as the Agent. The only difference is that the Agent ferries Ad-Requests and adverts, whereas the Commissioner ferries Ad-Reports. At this point we must note that **the terms Agent and Commissioner are designations that are given to a user based on the type of service that they provide**. Consider an example where a user named Alice sends an Ad-Request to Bob who transfers it to an Ad-Dealer, collects the advert and delivers it back to Alice. Alice then views the advert, generates an Ad-Report that she sends to Charlie who forwards it to the Ad-Dealer. In this particular scenario, the Agent is Bob and Charlie is the

Commissioner. Let us now examine a different scenario where Alice sends her Ad-Request to Danna who collects the adverts for Alice and then Alice composes her Ad-Report that is also sent to Danna who will deliver it to the Ad-Dealer. In this scenario, Danna serves both as Agent and Commissioner. It is clear that there is no practical restriction in regards to who serves as Agent or Commissioner and both responsibilities may be assumed by the same person. As to avoid any confusion, let it therefore be noted that **the term Agent is a designation that is given to the person who delivered an advert to the user and the term Commissioner is a designation that is given to the person that delivered the Ad-Report to the Ad-Dealer and both of these designations may be attributed to the same person**.

4.2. Adversary Model

Our system regards the majority of users as honest since they have no immediate benefit nor the necessary technical knowledge as to undermine the integrity of the system. However, it is possible for an adversary to easily assume the identity of a user without exposing himself since the only precondition for joining the system is to download a mobile client without revealing any identifying information. The main adversary is any group of malicious Ad-Dealers and Publishers who may attempt to commit fraud against the Advertisers by impersonating the identity of a legitimate user in order to submit fictitious Ad-Reports. The Advertisers are therefore willing to accept the authenticity of an Ad-Report only after it has been verified by the Broker whom they consider trustworthy.

The Ad-Dealers and Publishers have no reason to doubt the integrity of the Broker but at the same time are cautious of him as he might alter the content of an Ad-Report as to deprive them of their reword. The Broker therefore serves as a secondary adversary for the Ad-Dealers and Publishers.

One last aspect that needs to be considered is the privacy of the users that needs to be preserved. For this reason, the user considers all other parties, including Agents and Commissioners as honest but curious adversaries and is reluctant to share any personal information that can be associated to his or her identity (Maintaining user privacy when requesting and receiving adverts through an Agent is a matter that we resolved in [17] and for that reason we only focus on the delivery of Ad-Reports. Regardless of this, we do contemplate for all privacy aspects throughout the entirely of the model and thus our evaluation includes a section where we examine how our original advert distribution system is effected).

4.3. Problem Formalization

Given set of Ad-Dealers D, set of Publishers P, set of Users U who submit set of Ad-Reports AR, it should be possible for a Broker b to identify a subset of fictitious reports $\widehat{AR} \subseteq AR$ but at the same time it should not be possible for the Broker b , any Ad-Dealer $d \in D$ or any Publisher $p \in P$ to uncover the identity any user $u \in U$ nor alter the content of any $AR_u \in AR$ without being exposed.

4.4. Evaluation Criteria

Having considered the details of the tackled problem in Section 4.3 and after taking into account the scope of our research in Section 1.1, 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.

- **Protection against fabricated reports** A malicious user should not be able to commit fraud against the Advertisers by submitting reports which do not correspond to real consumer activity. Fraud prevention is the primary aim of analogous systems and therefore constitutes our main criterion.
- **Report integrity** After an Ad-Report has been sent, it should not be possible for the Ad-Dealers, Commissioners or Broker to alter its content without being exposed. In contrast to other models, our system utilizes a decentralized architecture over insecure channels and should therefore guarantee the integrity of transmitted data.

- **User privacy:** At any given moment, the Broker, Advertisers, Ad-Dealers, Agents and Commissioners should not be individually capable of associating a user's advertising interests to his or her identity. User privacy is an aspect that does not necessarily fall within the scope of fraud prevention systems but is a major concern for our research and is therefore taken into consideration.
- **Reporting effectiveness:** Ad-Reports should include all the necessary information as to ensure that all of the participating stakeholders are able to claim their reward. The Broker should be able to ensure that each report is accounted only once and also there should exist a way for the user to confirm that his or her reports were delivered successfully. 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).

5. Detailed System Analysis

In the following sections we provide a detailed analysis of our system and offer insight into our approach at solving the problem of detecting fake Ad-Reports without the need for knowing the identity of the submitting user. Our approach is based on the notion of behavioural verification, according to which a legitimate user can be verified by his social behaviour. More specifically, when users view adverts on their devices, they generate **Ad-Reports** as featured in Section 5.1.1. At the same time the users also collect a series of **Tokens** when they perform certain social tasks such as visiting specific locations or interacting with other users as explained in Section 5.2. Both the Ad-Reports and the Tokens are composed into an **Ad-Report Chain (ARC)** which is a blockchain-inspired construction that is further analyzed in Section 5.1.2.

As the *ARC* contains both the user's Ad-Reports and Tokens, it can be used by the Broker to verify that the user who created a particular Ad-Report has the same social behaviour as a legitimate user. Furthermore, the Tokens work as time-stamps which allow the Broker to verify that the reports of an *ARC* were created at a paced rate and not in bulk. The *ARC* can then be shared with the Ad-Dealers, Publishers and Advertisers so that the service rewards can be claimed. Information among the system stakeholders is shared through the use of a digital database termed as the **Service Confirmation Board (SC-Board)** illustrated in Section 5.1.3. The *SC-Board* offers additional security mechanisms which allow the user to confirm that his or her reports have been delivered without exposing the user's identity but also enables the Ad-Dealers and Publishers to verify that a submitted *ARC* has not been tampered by the Broker before being published.

5.1. Information Components

In the following sections we introduce the individual informational elements that compose our advert reporting system. More specifically, in Section 5.1.1 we specify the contents of the different types of Ad-Reports and in Section 5.1.2 we outline the way upon which the individual reports of a particular user can be combined into a blockchain-inspired construction that we label the Ad-Report Chain *(ARC)*. Finally, in Section 5.1.3 we present the concept of the Service Confirmation Board *(SC-Board)* that is a medium upon which *ARCs* may be shared with the separate stakeholders of the system.

5.1.1. Ad-Reports

The marketing industry generates profit from online advertising based mainly on three supported revenue models. The first and most popular is called **Cost-Per-Click (CPC)** advertising or Pay-Per-Click (PPC) and is founded on the principle that a reward is attributed when a user clicks on an advert that is displayed on the Publisher's domain. The second most prominent model is referred as **Cost-Per-Impression (CPM)** advertising or Pay-Per-Impression (PPM) and awards money when a user simply views a displayed advert. The third revenue model is known as **Cost-Per-Action (CPA)** or Pay-Per-Action (PPA) where money is awarded when users perform a specific action which is most typically the purchase of a product [18]. The preferences of Advertisers may not be limited to just a single pricing model and for this reason our system can support all of them at the same time by offering three different types of Ad-Reports as listed below.

- **RoV:** Report of View
- **RoC:** Report of Click
- **RoA:** Report of Action

As depicted in Figure 3, all supported Ad-Report types incorporate a sequence number N which indicates the order in which the reports were created. The *Advert Code* is a unique reference number that is sent to the user alongside each advert and can be used for identification. The D_{ID} and P_{ID} respectively accommodate the identities of the Ad-Dealer who distributed the advert and the Publisher who featured it to the user while the *Date* field holds the date and time of the publication.

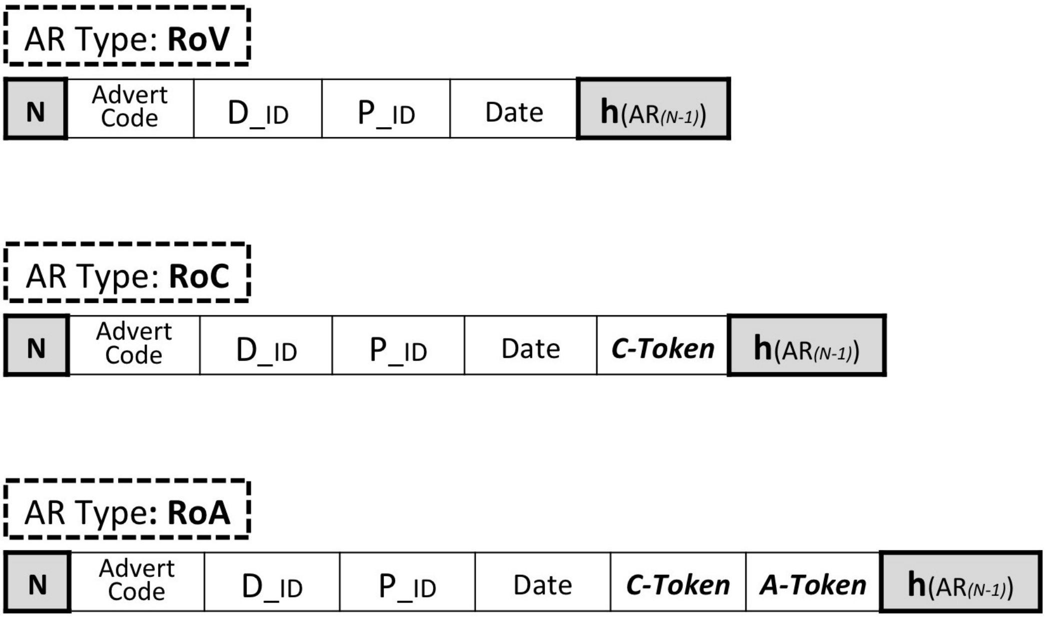

Figure 3. Supported types of Ad-Reports and their contents.

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 the advert is 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 the user only after a specific condition has been met (e.g., the user made a purchase or created an account). Each Advertiser periodically generates his own *C-Token* and *A-Token* which are uploaded within his domain. The function which is used for this operation as well as the frequency upon which the two tokens are updated fall udder the responsibility of the respective Advertisers. Ideally, the *C-Token* and *A-Token* should be generated by a cryptographically secure random number generator and as often as practically possible. The rate at which the *C-Token* and *A-Token* are updated influences the system's accuracy of verifying the time that the Ad-Reports were created. More specifically, if the Tokens are updated once every T time units, then the system can verify the time of a user's report with granularity T. It is assumed that the random number generator is secure and that the only feasible way to obtain the *C-Token* and *A-Token* is by downloading them from the locations in which they were uploaded by the particular Advertiser.

More specifically, the *C-Token* is uploaded in the same cuber-space 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 he or she performs 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 the Advertisers to verify that a user accessed their website or performed a specific action before creating a *RoC* or *RoA*. Having to obtain the 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 visit an advertiser's web site and to forge a *RoA* requires him to perform an action. More importantly, tokens prevent dishonest users from creating fictitious reports ahead of time as a *RoC* which contains a token C_n could not have been created before the token was made available online. Lastly, $h(AR_{(N-1)})$ contains a hash function digest of each previous Ad-Report that was composed by the same user. This enables the user to link all of the reports that he or she creates in the form of a blockchain-inspired architecture which is analyzed more minutely in Section 5.1.2.

One last thing that needs to be mentioned is the fact that for 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 a public key B_{PuK} which belongs to the Broker and is further clarified in Section 5.3.3.

5.1.2. Ad-Report Chain (ARC) and Integrity Hash (IH)

Rather than dealing with Ad-Reports individually as they are being created, our system enables each user to aggregate multiple Ad-Reports throughout the course of defined period and then submit all of them as a single unit. As it has already been explained in Section 5.1.1, each Ad-Report contains the hash digest of the previous. This enables the user to link several Ad-Reports together in a form that resembles the architecture of a blockchain which is termed the Ad-Report Chain or *ARC* for short.

As shown in Figure 4, the first block of the *ARC* contains an initiating value which is marked as *ARC-ID*. This 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 4.1 that the *RF* (Report Form) is a message that comes as a response to the user's request to file Ad-Reports.

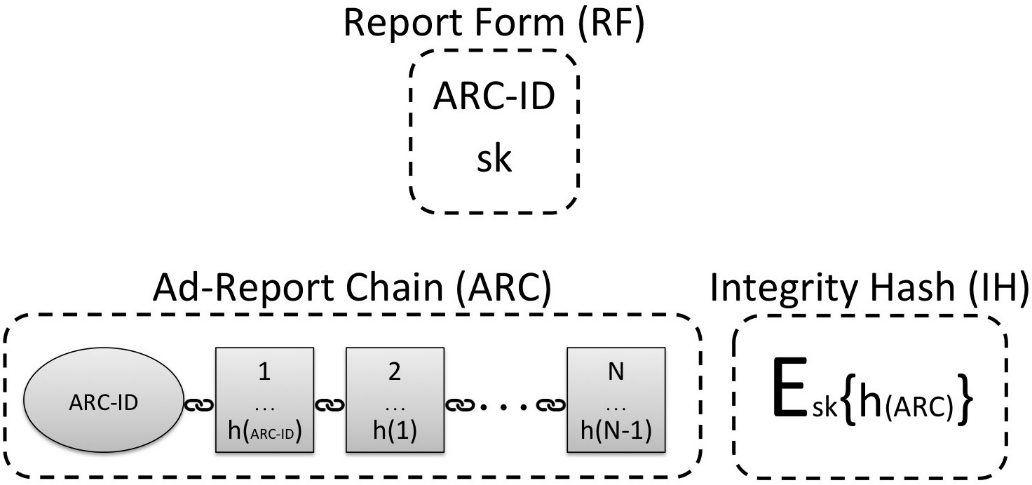

Figure 4. Structural elements of the Ad-Report Chain.

As we can see, the *RF* also contains a cryptographic signing key *sk*. While the *ARC-ID* is used to identify and mark the start of an *ARC*, the *sk* is used to mark the end in such a way that it prevents the removal of addition of blocks. More specifically, when the user completes the creation of the *ARC*, the user produces $h(ARC)$ and then signs it with *sk*. The resulting message, which we can also see in the same figure, is termed as the *IH* (or Integrity Hash) and is sent to the Broker along side the *ARC*.

The Broker can use a secret verification key vk in order to confirm that the *IH* was created by the user and then determine that the *ARC* has not been altered by comparing the $h(ARC)$ from within the *IH* to an $h'(ARC)$ which the Broker computes himself.

5.1.3. Service Confirmation Board (SC-Board)

The Service Confirmation Board (or *SC-Board* for short) is a digital database which serves as an information sharing platform between all of the system's stakeholders. The indexed entries of the *SC-Board* represent Request Forms that have been distributed to users and consist of five fields as shown in Figure 5.

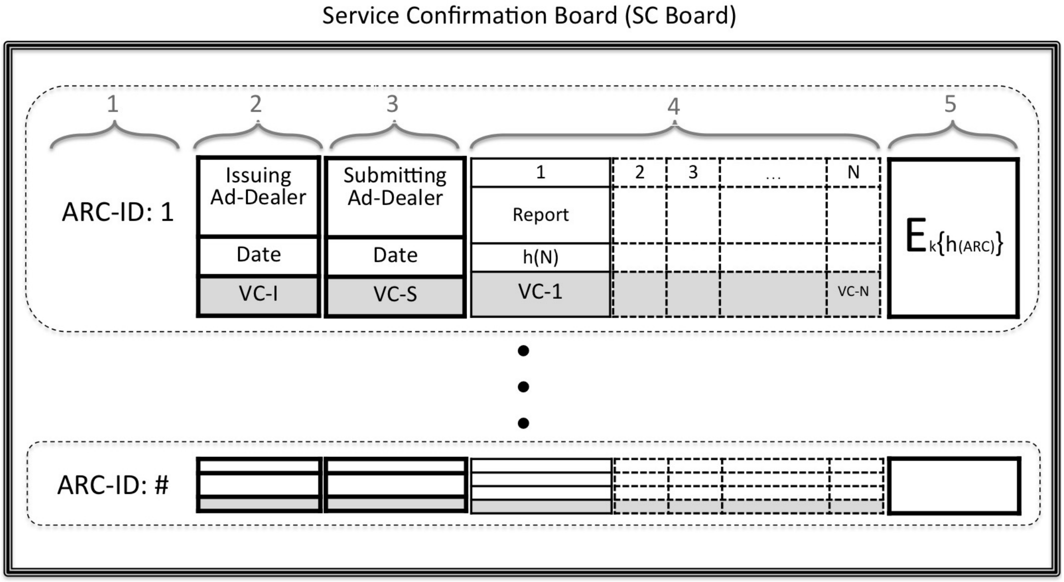

Figure 5. Service Confirmation Board architecture.

The first two fields are filled by the Broker when he issues a new *RF* and respectively contain the *ARC-ID* and the name 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 report is submitted with the third field keeping the identity of the submitting Ad-Dealer (the Ad-Dealer who forwarded the user's *ARC* and *IH*) and the date of submission while the forth 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 $h(N)$ for each block of the *ARC*. Recall that in Section 5.1.1, we briefly mentioned that the content of the Ad-Reports are 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 the Advertiser has awarded a commission to the Publisher. More details on the exact operation and the reasons behind it are provided in Sections 5.3.3 and 6. Lastly, we need to mention that all fields of the *SC-Board* are visible to Ad-Dealers, Publishers and Advertisers but the first field with the *ARC-ID* also becomes available to the users after submission has been completed. The 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

5.2. 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 on a large scale 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 period. While they are composing their Ad-Reports, honest users engage in typical social activities such as purchasing goods online, moving through space and interacting with other users. All of these social activities are distinguishing behaviours of honest users which can be exploited to verify their honesty without exposing their identity.

As we already described in Section 5.1.2, the 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 the Ad-Reports were being created.

5.2.1. 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, the system regards the engagement of 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, our system exploits other forms of honest behaviour as explained in the following sections.

In Section 5.1.1, 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 they contain tokens which are acquired by accessing the Advertiser's website. The *RoCs* and *RoAs* can therefore serve as indicators of honesty as they signify that the user took the time to visit the Advertiser's website. The remaining *RoVs* may not verifiable but they can be validated-by-association since the *ARC* follows the same architecture as a blockchain as shown in Figure 6.

A limitation to this approach lies in the fact that *RoCs* and *RoAs* are designed to be used by Advertisers who support the Cost-Per-Click (CPC) and Cost-Per-Action (CPA) pricing models. This may limit the number of *RoCs* and *RoAs* as it excludes all the Advertisers who only support Cost-Per-Impression (CPM). To overcome this shortcoming, our system utilizes the different types of Ad-Reports (*RoV*, *RoC* and *RoA*) not based on the Advertiser's pricing model but in accordance to the user's engagement with the advert. Regard a simple example where an Advertiser supports the CPM model 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 making a purchase. The commission is still going to be awarded based on the viewing but the use of a more secure Ad-Report will validate the authenticity of the claim as *RoVs* can be forged more easily than *RoCs* and *RoAs*.

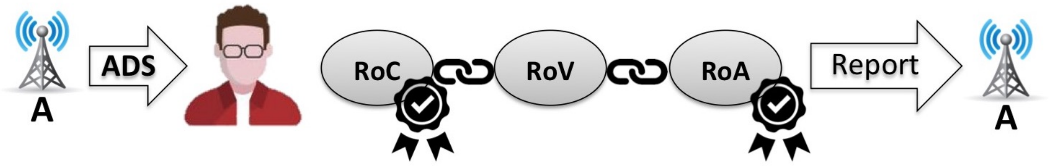

Figure 6. Behavioural verification by advert association.

5.2.2. Time and Location Checkpoint

The use of *RoCs* and *RoAs* enables the identification of honest users in two ways: First, they are indicators of a user who took the time to perform a specific action; second, the *C-Token* and *A-Token* (which are periodically updated) can be used to determine the rate at which the Ad-Reports were created. 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 would signify the time that a particular block was created. One limitation that needs to be considered however, is that this *T-Token* should not 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. The correctness of the *T-Token* is ensured through the synchronization of Ad-Dealers which can easily be achieved with the help of the Broker. This enables the verification of 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 [19]. The location tag provides additional proof of the user's honesty as it verifies the user's social behaviour in terms of appearing within the proximity of public locations, where Ad-Dealers are broadcasting, such as shopping malls, cafes and WiFi hotspots.

To further comprehend this notion, consider the following example of attempted forgery. If the *T-Token* were accessible online, a dishonest user $\widehat{u}$ could periodically download it and use it to easily verify set $\widehat{S} = \{\widehat{ARC}_1, \widehat{ARC}_2, ..., \widehat{ARC}_i\}$ of fictitious *ARCs* over a longer period of time. However, when the *T-Token* is distributed by the Ad-Dealers, it is more difficult for $\widehat{u}$ to obtain it since he or she needs to physically travel to the location of the Ad-Dealer and request multiple copies for each of the elements of $\widehat{S}$. 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 the adversary.

The *T-Token* is obtained and embedded into a user's *ARC* as follows. When entering the vicinity of an Ad-Dealer, the user sends the hash digest $h(n-1)$ of the last block in the *ARC* and a user-generated encryption key U_{EK}, both encrypted with the Ad-Dealer's public key D_{PuK}. The Ad-Dealer decrypts the message with his private key D_{PrK} and composes a Checkpoint Block (*CB*). As illustrated in Figure 7, the *CB* contains the identity *B* who is the Ad-Dealer that performed the check as well as the received hash $h(n-1)$ and date which are signed with a private key *KB* that belongs to him and is only used for this application. The date serves as the *T-Token* while the signature is proof of the user's location. Before being sent back to the user, the *CB* is first encrypted with the Broker's public key B_{PuK} and then encrypted again with the user's U_{EK} as shown in Equation (1):

$$E_{U_{EK}}[E_{B_{PuK}}[CB]] \tag{1}$$

When the cryptogram is received, the user decrypts it with the corresponding U_{DK} and obtains the $E_{B_{PuK}}[CB]$ which is given a sequence number $N = n$ and is inserted into the user's *ARC* as shown in Figure 7 (All blocks of the *ARC* are encrypted with the Broker's public key B_{PuK} as to ensure privacy against Ad-Dealers and Commissioners. The encryption on the *CB* could had been performed by the user himself but in this case is performed by the Ad-Dealer in order to relieve some of the strain from the user's mobile device. The Ad-Dealer can be trusted with this operation as he has no benefit from providing a defective *CB*).

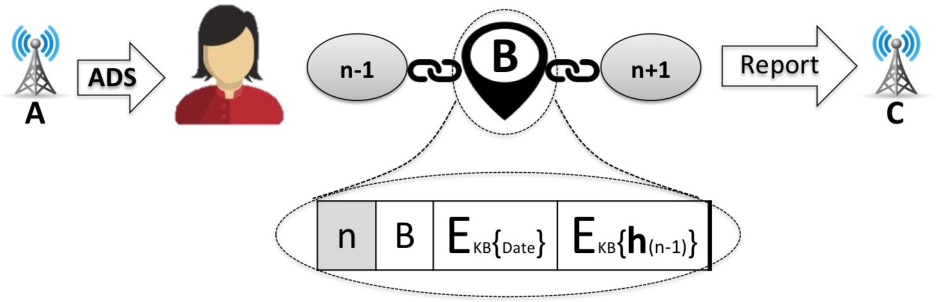

Figure 7. Behavioural verification by checkpoint.

5.2.3. 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 the sequence numbers *N* and hashes $h(n)$ of their last blocks as well as their *ARC-IDs*. The two users can then verify the date and time of the meeting by adding an Affiliation Block *AB* in their respective *ARCs* with each others' information. In order to be valid, the *ABs* which are added to the *ARCs* of both users need to have matching dates and times but this does not require a perfect synchronization between the two users. Mobile applications typically have a recommended refresh rate for adverts that is between 30 s to 120 s while malicious applications generate Ad-Reports at a 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 *ABs* do not match because one of the users provided an unrealistically 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 8 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 **Date** of the meeting (as registered by *B*) and the information that was sent by *A* which includes her *ARC-ID=ARC1* as well as the sequence number *m* and digest $h(m)$ of her last block. The date which is added by *B* works as a *T-Token* which verifies the last block of *ARC1* at a particular time. Notice that *ARC1* and sequence number *m* are sent encrypted with the Broker's public key B_{PuK} while the $h(m)$ is signed with *A*'s signing key *sk1* which is also used for the creation of the Integrity Hash *IH* as explained in Section 5.1.2. This ensures that *B* does not lean any information about *ARC1* and is not able to alter $h(m)$.

Through the exchange of *ABs*, the Broker can infer that two *ARCs* were submitted by affiliated users but this does not compromise user privacy in any way. The *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. One limitation of this verification method lies on the fact that a dishonest user may exchange *ABs* between multiple fictitious *ARCs*. Although plausible, this is prevented by combining all three verification methods as discussed in the following Section 5.2.4.

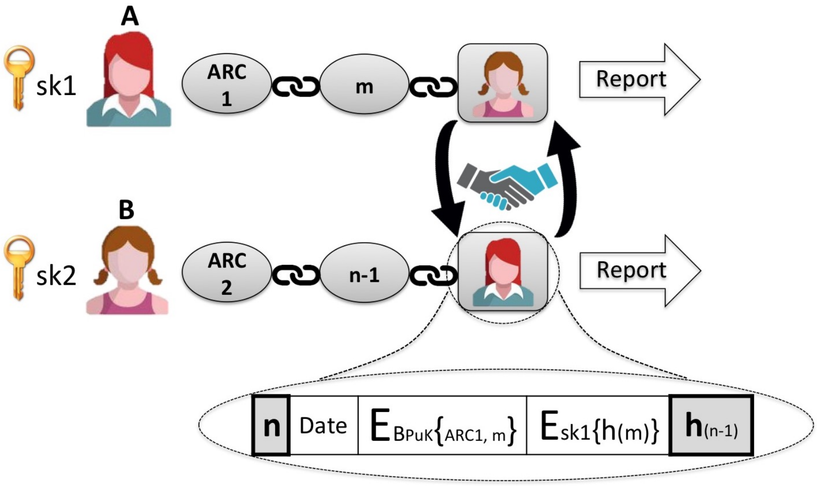

Figure 8. Behavioural verification by social affiliation.

5.2.4. Combined Verification

The individual methods of behavioural verification have certain limitations as the social affiliation approach in Section 5.2.3 is susceptible to fraud by means of creating multiple fictitious *ARCs* while the methods which are described in Sections 5.2.1 and 5.2.2 may not always be practically feasible as they require the 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 9, 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 from an Ad-Dealer and two Affiliation Blocks.

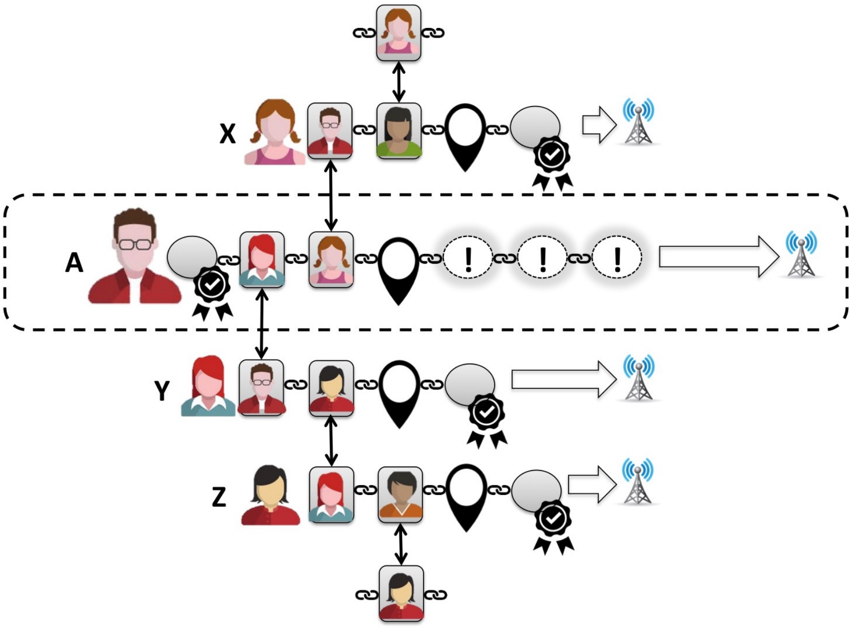

Figure 9. Extended diagram of behavioural verification.

Furthermore, we see that the respective *ARCs* of the two users *X* and *Y* who provided *ABs* for *A* also have verifiable reports, *CBs* and *AB* from other users such as *Z*. 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 example 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.

5.3. Protocol Description

The system comprises of three sub-protocols which correspond to the three stages of the system's operation (1) Report Form Collection, (2) AD Collection and (3) Ad Report Submission that were mentioned in Section 4.1. The sub-protocols run sequentially and are detailed in the following sections.

5.3.1. Report Form Collection Sub-Protocol

The Report Form Collection sub-protocol that is depicted in Figure 10 is run when the user needs to acquire a new Report Form.

1. The user calculates a pair of asymmetric keys U_{EK} and U_{DK}. The encryption key U_{EK} is composed into an *RF-Request* and the decrypton key U_{DK} is temporarily stored.
2. The *RF-Request* is encrypted with the Broker's public key B_{PuK} and sent to one of the Ad-Dealers, either directly or via an Agent.
3. Upon receiving the *RF-Request*, the Ad-Dealer forwards it to the Broker.
4. The Broker decrypts the *RF-Request* with his private key B_{PrK} and obtains the user's encryption key U_{EK}. The Broker then creates an entry in the *SC-Board* with a new *ARC-ID* and the identity of the issuing Ad-Dealer and afterwards computes a pair of keys *sk* and *vk*. The Verification Key *vk* is stored securely while the Signing Key *sk* and the *ARC-ID* are composed into a Request Form *RF*.
5. The *RF* is first encrypted with the U_{EK} which was provided by the user and then is returned to the issuing Ad-Dealer who filed the request.
6. The **issuing** Ad-Dealer verifies his identity on the *SC-Board* by providing this signature.
7. The Ad-Dealer forwards the *RF* to the user either directly or via the same Agent who delivered the original *RF-Request*.
8. The user receives the encrypted *RF* and decrypts it with his U_{DK} in order to obtain the *ARC-ID* and *vk*.

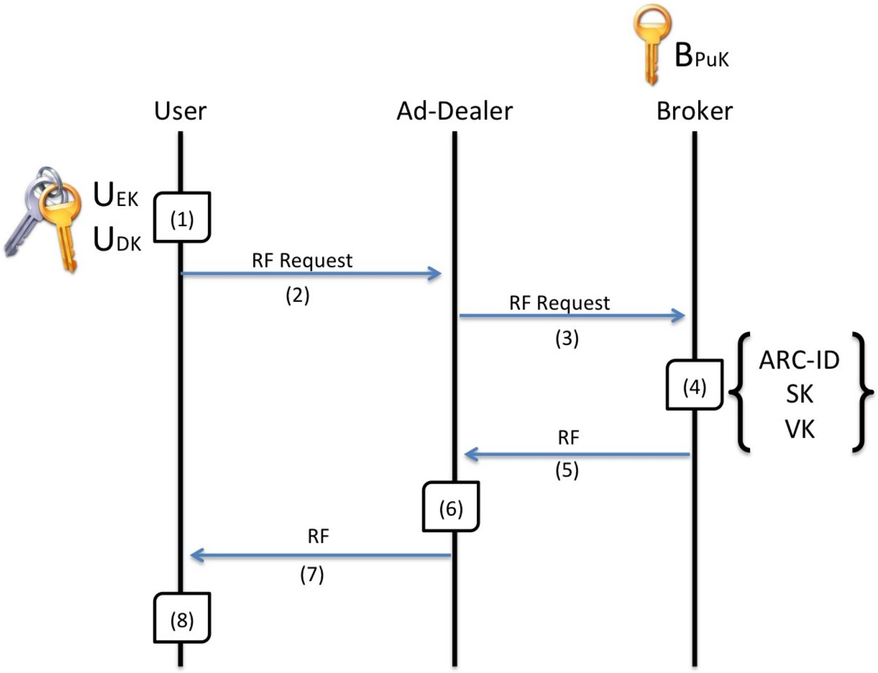

Figure 10. Report Form Collection sub-protocol.

5.3.2. Advert Collection Sub-Protocol

The Advert Collection sub-protocol that is depicted in Figure 11 is used for the acquisition of adverts and may run multiple times as needed.

1. The user computes a pair of asymmetric keys U'_{EK} and U'_{DK}. The user stores the decryption key U'_{DK} and then composes an Advert Request Message (*ARM*) which contains the encryption key U'_{EK} and his advertising interest (referenced by known interest identifies that are common throughout the system).
2. The *ARM* is encrypted with a public key D_{PuK} and sent to one of the Ad-Dealers, either directly or via an Agent.
3. The responding Ad-Dealer decrypts the *ARM* with the corresponding key D_{PrK} which is kept private among all the Ad-Dealers who use the system and obtains U'_{EK} as well as the user's advertising interest.
4. The Ad-Dealer determines which specific adverts would best suit the needs of the user and files a request to the appropriate Advertisers.
5. Upon receiving the Ad-Dealer's request, the Advertisers respond with the adverts in plain form.
6. The Ad-Dealer aggregates all of the received adverts into an Advert Delivery Message *ADM*.
7. The *ADM* is first encrypted with the U'_{EK} and then returned to the user either directly or via the same Agent who delivered the original *ARM*.
8. The user receives the decrypts the *ADM* with his U'_{DK} in order to obtain the adverts which are stored for future use by the Publishers.

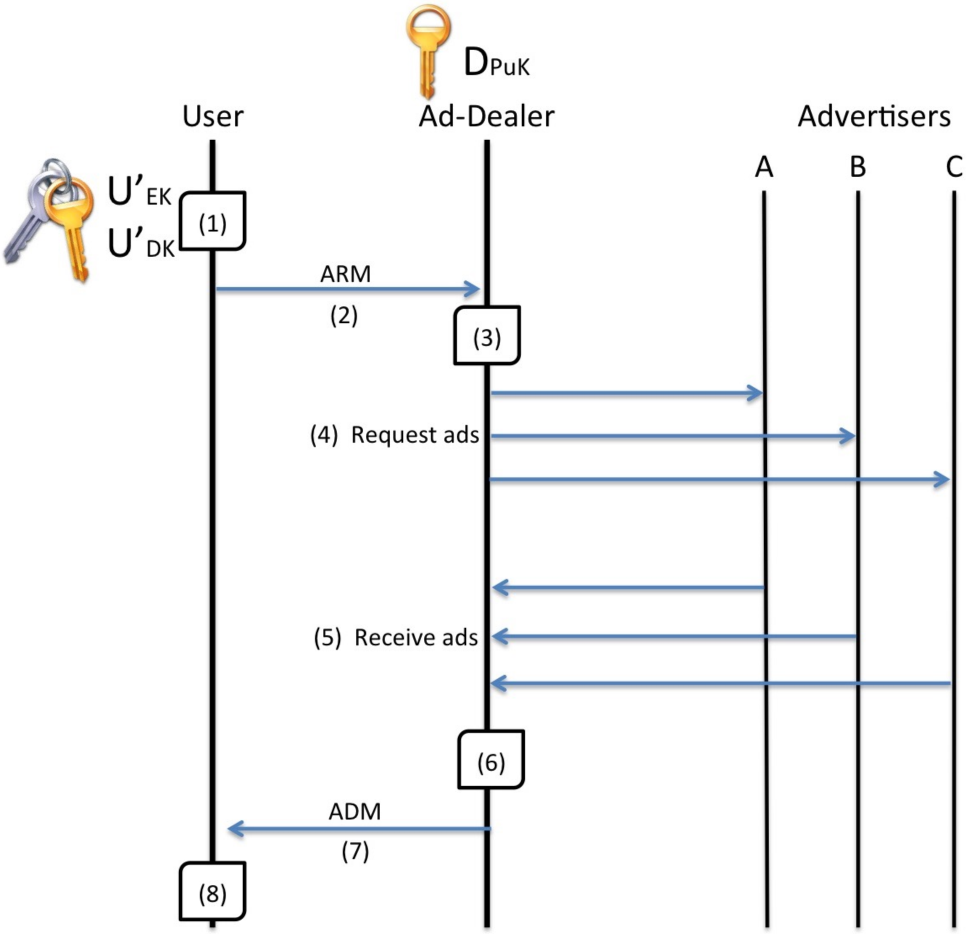

Figure 11. Advert Collection sub-protocol.

5.3.3. Ad Report Submission Sub-Protocol

The Advert Collection sub-protocol that is depicted in Figure 12 is used for the delivery of Advert Reports by the user to the Broker.

1. The user gradually composes his or her reports into an *ARC* as he or she is interacting with adverts. The contents of the *ARC* are encrypted with the Broker's public key B_{PuK} except for the first 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 signing the result with the signing key *sk* which were delivered to him with the sub-protocol described in Section 5.3.1.
2. The *ARC* and *IH* are sent to one of the Ad-Dealers either directly or via a Commissioner.
3. The Ad-Dealer keeps a local copy of the *ARC* and *IH*.
4. The Ad-Dealer submits the *ARC* and *IH* to the Broker.
5. The Broker first decrypts the *ARC* with his private key B_{PrK} and verifies the authenticity of the *IH* with the matching verification key *vk*. 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*. The *CBs* are validated by checking the authenticity of the Ad-Dealers signature with a public key which is used only for this application. The *ABs* are validated by checking the signature (which exploits the *sk′* of the second user who participated in the meeting) and by cross referencing the dates in both the *ARCs* (if the other *ARC* has already been submitted).
6. When the *ARC* and *IH* have been uploaded to the *SC-Board*, the Broker notifies the **submitting** Ad-Dealer with a Check Message.
7. 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 originally encrypted). 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*.
8. The Broken notifies the Advertisers for the new entry in the *SC-Board*.
9. The Advertisers begin to reward the Publishers and Ad-Dealers and each reward report of the *ARC* is marked on the *SC-Board* by the appropriate Advertiser. Each Advertiser is responsible for individually determining the honesty of the user who submitted the *ARC* 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 conforming the contained *C-Tokens* and *A-Tokens*. Depending on the number and significance of credentials in the *ARC*, the Advertisers 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*).
10. 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 his or her report has been registered. If the matching *ARC-ID* is present within the *SC-Board*, the user may discard his or her original copy of the *ARC* and *IH* or else resubmits 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.

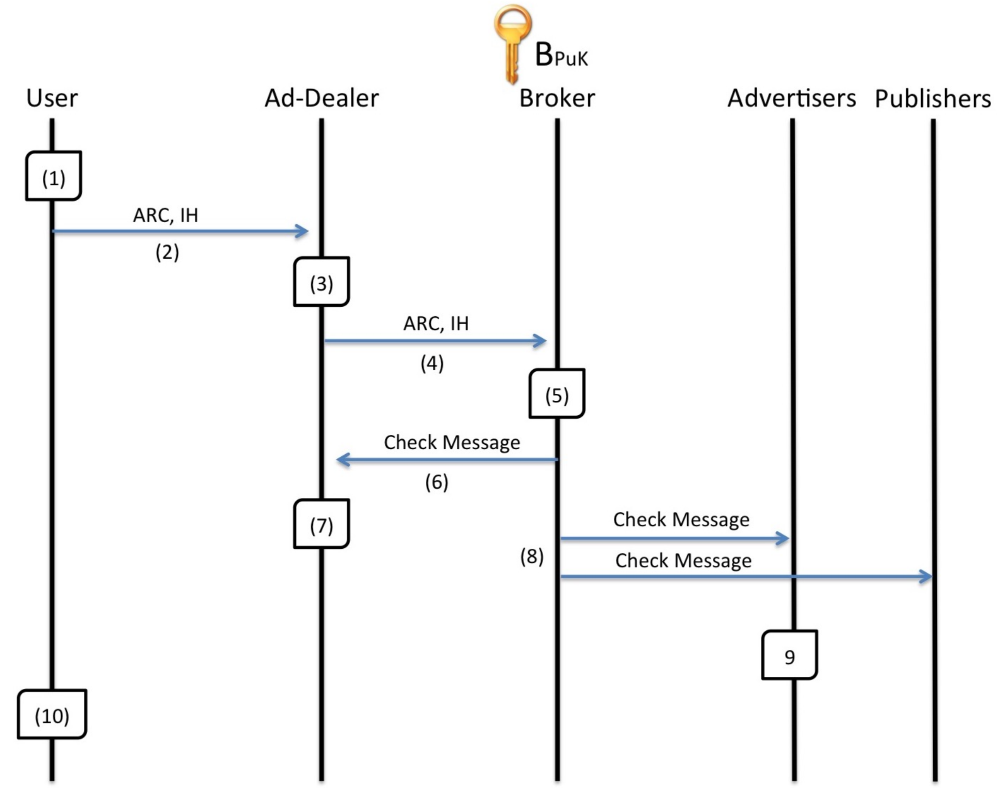

Figure 12. Ad Report Submission sub-protocol.

6. Evaluation

We follow a qualitative approach to evaluate our work by scrutinizing the performance of our system against the evaluation criteria that we established in Section 4.4. The goals are (1) to determine the effectiveness of our model at delivering accurate advert reports, (2) assess the system's resilience to fraudulent behaviour which is targeted against both the Advertisers as well as the Ad-Dealers and Publishers, and lastly (3) to evaluate the effectiveness of the system at preserving the privacy of the user.

6.1. Reporting Effectiveness

Reporting effectiveness is defined as the system's capability to deliver the necessary information which is required by the Publishers and Ad-Dealers (as a substitute of the Ad-Network) to claim their rewards from the Advertisers. As we do not process the data of Ad-Reports beyond encrypting and decrypting it, the information delivered by our system is the same as in the currently deployed system. In that regard, our system offers a similar reporting effectiveness in terms of the quality of information which it supports. In addition to this, our approach allows for the acquisition of supplementary information about the social habits of consumers. By matching the *ABs* (Affiliation Blocks) within the *ARCs* (Ad-Report Chain), the Broker would be able to deduce the consumer interest similarity of users that belong in the same social cycle. Furthermore, the Broker can also infer correlations between adverts, Publishers and locations (e.g., users who view adverts of product A are also interested in product B or users who visit location X tend to view adverts through Publisher Y). This offers a very useful insight into consumer practises which is obtained without violating the user's privacy by tracking his or her IP address.

As *ARCs* are shared on the *SC-Board* under specific *ARC-IDs*. This ensures that all stakeholders have access to them and that the same Ad-Report is not rewarded more than once. Moreover, the *SC-Board* allows partial access to users which enables them to detect when an *ARC* has not been delivered on time and may need to be resent, thus increasing the system's robustness. Although participation is voluntary, the privacy that the system provides is a good incentive for attracting users.

6.2. Fraud Protection of Advertisers

Financial fraud against the Advertisers is the main shortcoming of the currently enforced model that we address with our work. 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, our model enables the affected stakeholders (Advertisers, Publishers and Broker) to (1) calculate the rate upon which a user creates reports and (2) verify that a user has the same behavioural patterns as a typical consumer.

To illustrate the system's protection against fraud, we will examine an attack scenario where a dishonest user $\widehat{U}$ attempts to commit fraud at a large scale by submitting a set $\widehat{S} = \{\widehat{ARC}_1, ..., \widehat{ARC}_i\}$ of fictitious *ARCs*. Recall that an *ARC* contains 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 reports, the *RoV* is the easiest to fabricate as it does not contain any verifiable information (*Tokens*). However, $\widehat{U}$ cannot submit an $\widehat{ARC}$ which only contains *RoVs* as this would be immediately rejected by the Broker. The *RoC* and *RoA* contain a *C-Token* (Click-Token) or an *A-Token* (Action-Token) which can only be obtained by visiting the Advertiser's website within a particular time-frame. This prevents $\widehat{U}$ from creating *RoCs* and *RoAs* ahead of time but $\widehat{U}$ can still attempt to commit fraud by creating the $\widehat{ARC}$ over a longer period of time. Although this makes the creation of the $\widehat{ARC}$ more difficult, it is still possible with the use of an automated process that automatically downloads the *Tokens* when they become available. Despite this fact, the $\widehat{ARC}$ would still be rejected by the Broker as it would not contain any *CBs* or *ABs*.

Recall that *CBs* are distributed by Ad-Dealers and include a *T-Token* (Time-Token) which is signed and cannot be forged without knowledge of a secret cryptographic key. In order to obtain valid *CBs*, $\widehat{U}$ would need to repeatedly travel to the physical location of an Ad-Dealer throughout the course of the creation possess of the $\widehat{ARC}$. Moreover, the fraudster must be cautious not to request multiple *CBs* (for different $\widehat{ARCs}$) at the same time as this would provoke suspicion. Even if $\widehat{U}$ were to conspire with one of the Ad-Dealers, the $\widehat{ARC}$ 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, the fraudster would need to conspire with multiple Ad-Dealers and manage the *CBs* in a way that does not create an observable pattern (e.g., multiple $\widehat{ARCs}$ containing *CBs* from the same group of Ad-Dealers). Provided that the Ad-Dealers are carefully selected, such a scenario would be unlike.

The *ABs* are exchanged between users and serve a similar purpose as *CBs* as they can be used to determine the rate in which the reports 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 $\widehat{U}$ can exchange *ABs* between multiple fake $\widehat{ARCs}$. However, if the fraudster was to compose a $\widehat{ARCs}$ 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 $\widehat{U}$ to fabricate $\widehat{ARCs}$ which are realistic enough to fool the Broker, $\widehat{U}$ would need to use an automated process which downloads *C-Tokens* and *A-Tokens* over an extended period of time. During that time, $\widehat{U}$ would need to exchange *ABs* between the $\widehat{ARCs}$ and also physically collect *CBs* from different Ad-Dealers without raising their suspicion by submitting multiple requests at the same time. This process is time consuming and impractical which makes the conduct of financial fraud more difficult to accomplish.

6.3. Report Integrity

After an *ARC* leaves the user's device, it has to go through a Commissioner, 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 an *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*. Our system 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 Commissioner 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 includes the hash digest of the previous. Additionally, the user also sends an *IH* (Integrity Hash) that contains the hash digest of the entire *ARC* which has been signed with a verification key *vk*.

Since the content of the Ad-Reports is encrypted with the Broker's public key B_{PuK}, it would be possible for a malicious Commissioner or Ad-Dealer to create a fictitious Ad-Report of his own. 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 as well as the hash digest that is included in the *IH*. The attacker would be able to change the hashes in the *ARC* but cannot change the hash in the *IH* without knowing the user's verification key *vk* which was delivered to the user within the *RF* (Report Form). The *RF* was encrypted with a key that the user selected himself and only shared with the Broker via his public key B_{PuK}. This makes it impossible for an attacker to alter the content of *IH* without being exposed.

Attack scenario 2:

The Broker attempts to alter the content of a legitimate *ARC* in order to cheat a Publisher out of a reward. Although the Broker can be considered trustworthy as the representative of Advertisers, our system allows the Ad-Dealers and Publishers to verify that there has been no tampering of the Ad-Reports.

Recall that Ad-Reports are encrypted with the Broker's public key B_{PuK} and are shared on the *SC-Board* (Service Confirmation Board) after they have been decrypted. The Broker could therefore attempt to cheat the 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 holds a copy of the *ARC* before forwarding it to the Broker. Although certain parts of the *ARC* are encrypted with Broker's public key B_{PuK}, the hash digests are transferred in plaintext. This allows the submitting Ad-Dealer to replicate the hash functions on the posted *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. User Privacy

User privacy is ensured through the application of encryption, anonymous connections and the employment of Agents and Commissioners who serve the role of partially trusted proxies. The Advert Request Message (*ARM*) which is sent by the user to request adverts is encrypted with the Broker's public key B_{PuK}. The requested adverts are placed in an Advert Delivery Message (*ADM*) that is encrypted with an encryption key U'_{EK} which was provided by the user within the *ARM*. This ensures that the Agent who serves the user has no access to neither the requests nor the ads. Similarly, the Ad-Reports of the user's *ARC* are also encrypted with the Broker's public key B_{PuK} which ensures privacy against the serving Commissioner. The Broker does not learn the identities of the user, the Agent or the Commissioner as the exchange of information takes place over an anonymous connection. The same anonymous connection is also in place when the user collects a Checkpoint Block (CB) from an Ad-Dealer within his or her proximity. The request is encrypted with the Ad-Dealer's public key D_{PuK} and contains a user-generated encryption key U_{EK} which is used by the Ad-Dealer to transfer the CB. This ensures that an eavesdropper cannot gain access to the user's information nor spoof the identity of the Ad-Dealer in order to transmit a fake CB.

7. Conclusions

Digital advertising allows Advertisers to effectively target users with promotional materials and at the same time offers a sustainable source of income for Publishers. Advertising fraud however, presents a serious threat for the marketing industry as it is costing Advertisers billions of dollars every year. The attempts of Ad-Networks to contain the problem by filtering incoming traffic has had limited effectiveness against BotNets and Click-farms who employ various methods in order to remain undetected.

In this paper, we address the problem by contributing an alternative reporting method which filters Ad-Reports by determining if they were composed by honest or dishonest users. What constitutes a user's honesty is the rate at which they create Ad-Reports. While honest users view adverts at a paced manner over a longer period, dishonest users (whether BotNets or Click-farms) generate large volumes of reports in a short amount of time. The challenge that we face is the need to protect the user's anonymity which prevented us from simply verifying real users through traditional means such as digital signatures. To overcome this limitation, our approach identifies honest users based on their behavioural patterns. Typical social behaviours such as making online purchases, traveling to specific locations and engaging with other users are exploited by our system in order to allow users to collect a series of digital credentials which we call *Tokens*. The *Tokens* work as proof that the user who collected them was engaged in a specific social activity (e.g., traveling to designated location) at a particular time. When combined with Ad-Reports in the form of an *ARC* (Ad-Report Chain) which follows the same architecture as that of a blockchain, the *Tokens* operate both as time-stamps which give us the ability to determine the rate the reports were created at but also offer proof that the creator has the type of social activity which is indicative of an honest user. The *Tokens* do not include any identifying information and the *ARC* is then delivered through the use of opportunistic networking and anonymous connections which ensure the user's privacy. This enables Advertisers to validate multiple reports which are composed by the same user without the user's identity being exposed. The combination of effective Ad-Report verification and user anonymity offers a well-balanced solution where both the Advertisers and users achieve their goals without one of them needing to make a compromise. Our method of verifying users based on their behaviour creates a promising alternative to verification methods which involve tracking and therefore violate the user's privacy.

Author Contributions: Stylianos S. Mamais, System design and manuscript creation; George Theodorakopoulos, Corrections and suggestions.

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

References

1. Digital Ad Spending to Surpass TV Next Year-eMarketer. 2016. Available online: https://www.emarketer.com/Article/Digital-Ad-Spending-Surpass-TV-Next-Year/1013671 (Accessed on 10 May 2017).
2. What Is an Untrustworthy Supply Chain Costing the US Digital Advertising Industry? 2015. Available online: http://www.iab.com/wp-content/uploads/2015/11/IAB_EY_Report.pdf (Accessed on 11 May 2017).
3. Mungamuru, B.; Weis, S.; Garcia-Molina, H. *Should ad Networks Bother Fighting Click Fraud? (Yes, They Should)*; Technical Report; Stanford InfoLab Publication Server, Stanford, CA, USA , 2008.

4. Vidros, S.; Kolias, C.; Kambourakis, G.; Akoglu, L. Automatic detection of online recruitment frauds: Characteristics, methods, and a public dataset. *Future Internet* **2017**, *9*, 6.
5. Walgampaya, C.; Kantardzic, M.; Yampolskiy, R. Real time click fraud prevention using multi-level data fusion. In Proceedings of the World Congress on Engineering and Computer Science, San Francisco, CA, USA, 20–22 October 2010; Volume 1, pp. 20–22.
6. Juels, A.; Stamm, S.; Jakobsson, M. Combating click fraud via premium clicks. In Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium, Boston, MA, USA, 6–10 August 2007; USENIX Association: Berkeley, CA, USA, 2007; pp. 2:1–2:10.
7. Haddadi, H. Fighting online click-fraud using bluff ads. *ACM SIGCOMM Comput. Commun. Rev.* **2010**, *40*, 21–25.
8. Iqbal, M.S.; Zulkernine, M.; Jaafar, F.; Gu, Y. Fcfraud: Fighting click-fraud from the user side. In Proceedings of the 2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE), Orlando, FL, USA, 7–9 January 2016; pp. 157–164.
9. Faou, M.; Lemay, A.; Décary-Hétu, D.; Calvet, J.; Labrèche, F.; Jean, M.; Dupont, B.; Fernande, J.M. Follow the traffic: Stopping click fraud by disrupting the value chain. In Proceedings of the 2016 IEEE 14th Annual Conference on Privacy, Security and Trust (PST), Auckland, New Zealand, 12–14 December 2016; pp. 464–476.
10. Pizzolante, R.; Carpentieri, B.; Castiglione, A.; Castiglione, A.; Palmieri, F. Text compression and encryption through smart devices for mobile communication. In Proceedings of the 2013 IEEE Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Taichung, Taiwan, 3–5 July 2013; pp. 672–677.
11. Papadimitratos, P.; Haas, Z.J. Secure routing for mobile ad hoc networks. In Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS), San Antonio, TX, USA, 27–31 January 2002; pp. 193–204.
12. Arapinis, M.; Mancini, L.I.; Ritter, E.; Ryan, M.D. Analysis of privacy in mobile telephony systems. *Int. J. Inf. Secur.* **2017**, *16*, 491–523.
13. Traynor, P.; Lin, M.; Ongtang, M.; Rao, V.; Jaeger, T.; McDaniel, P.; La Porta, T. On cellular botnets: Measuring the impact of malicious devices on a cellular network core. In Proceedings of the ACM 16th ACM Conference on Computer and Communications Security, Chicago, IL, USA, 9–13 November 2009; pp. 223–234.
14. Lilien, L.; Kamal, Z.H.; Bhuse, V.; Gupta, A. Opportunistic networks: The concept and research challenges in privacy and security. In Proceedings of the WSPWN, Miami, FL, USA, March, 2006; pp. 134–147.
15. Boldrini, C.; Conti, M.; Passarella, A. Exploiting users' social relations to forward data in opportunistic networks: The HiBOp solution. *Pervasive Mob. Comput.* **2008**, *4*, 633–657.
16. Theodorakopoulos, G.; Boudec, J.Y.L.; Baras, J.S. Selfish response to epidemic propagation. *IEEE Trans. Autom. Control* **2013**, *58*, 363–376.
17. Mamais, S.S.; Theodorakopoulos, G. Private and secure distribution of targeted advertisements to mobile phones. *Future Internet* **2017**, *9*, 16.
18. Fain, D.C.; Pedersen, J.O. Sponsored search: A brief history. *Bull. Am. Soc. Inf. Sci. Technol.* **2006**, *32*, 12–13.
19. Narayanan, A.; Thiagarajan, N.; Lakhani, M.; Hamburg, M.; Boneh, D. *Location Privacy via Private Proximity Testing*; NDSS; Stanford University: Stanford, CA, USA, 2011; Volume 11.