MetaPriv: Acting in Favor of Privacy on Social Media Platforms

Robert Cantaragiu[1], Antonis Michalas[1,2], Eugene Frimpong[1], and Alexandros Bakas[1]

[1] Tampere University, Finland
[2] University of Westminster, U.K.

Abstract. Social networks such as Facebook[3] and Instagram are known for tracking user online behaviour for their own commercial gain. To this day, there is practically no other way of achieving privacy in said platforms other than renouncing their use. However, many users are reluctant in doing so because of convenience or social and professional reasons. In this work, we propose a means of balancing convenience and privacy on Facebook through obfuscation. We have created `MetaPriv`, a tool based on simulating user interaction with Facebook. `MetaPriv` allows users to add noise to their account so as to lead Facebook's profiling algorithms astray and make them draw inaccurate profiles in relation to their interests and habits. To prove the tool's effectiveness, we ran extensive experiments over a 10-week period. Our results showed that privacy is achieved by fuddling Facebook, when the amount of traffic generated by the tool is similar to that generated by users on a regular basis. We believe that `MetaPriv` can be further developed to accommodate other social media platforms and help users regain their privacy, while maintaining a reasonable level of convenience. To this end and in view of supporting open science and reproducible research, our source code is available online.

Keywords: Metaverse, Obfuscation, Online Profiling, Privacy, Social Networks, Recommendation Systems

1 Introduction

In the past decades, online tracking on social networks has risen concerns regarding user privacy. Recommendation systems on social media platforms are developed to present biased information with the purpose of encouraging user engagement. When users indicate their opinions, beliefs and preferences on said platforms – whether by clicking 'like' on an article or by writing a controversial post – the recommendations they receive attempt to reinforce their beliefs. This aims at providing users with information that most likely interests them and enables them to trace other users sharing the same values. Through this approach,

[3] Since October 2021 is also known as META.

users gradually become more engaged in platforms, while going deeper in the rabbit-hole of subjectivity, since the only information and news they receive affirms their already established opinions. As a result, users remain engaged in social network platforms, as the latter make accurate predictions on their potential consumption needs. Hence, platforms in collaboration with companies promoting their products manipulate user information for targeted advertising.

Balance between Privacy and Convenience on Social Networks: Most users seem to be left with two options when it comes to social network privacy: *(1)* either regular use of the platform – hence no privacy or *(2)* complete abstinence from social networks – hence full privacy. However, the second option presents a number of problems. First, the hassle of removing data about oneself from a platform, discourages users as it demands tedious action. Note that data removal does not refer to deleting the account alone, but to the deletion of all posts, pictures and logged data from the platform. Secondly, even in cases where all user data is deleted, social networks may still track individuals through partner companies on different websites (e.g. through Facebook Pixel[4]). Finally, completely opting out of social networks results in great costs in terms of convenience for many individuals, who wish to keep in touch with their friends, keep up with the news and promote themselves or their activities. To this end, we believe that complete privacy is not achievable for most users. We do, however, think that one can strike a balance between privacy and convenience on said platforms and this has been a major motive behind our work. Our platform of choice for this work is Facebook – the world's largest online social network. However, the idea presented below can be developed to accommodate privacy on other platforms.

Contributions: The main idea in this work has been developed based on increasing concerns regarding the breach of user privacy in online social networks. More precisely, the main concern is that user choices are being covertly manipulated and controlled by social networks. With this in mind, we built `MetaPriv`, an automated tool that allows Facebook users to obfuscate their data and conceal their real interests and habits from Facebook. As a result, the core contribution of this paper is that it provides users with the necessary tools to protect their privacy when using social networks. It is worth mentioning that `MetaPriv` allows users to define the desired level of privacy (e.g. become almost 'invisible' online while still using social network platforms, reveal certain information about their digital and real lives etc.). By doing this, `MetaPriv` provides a novel and adaptive balance between privacy and functionality. This is a feature we believe will be used in several services in the near future.

Organization: The rest of the paper is organized as follows: In section 2, we present important published works on the topic of social network privacy. In section 3 we describe the components of our solution and give a high-level overview of `MetaPriv`. The assumed threat model is presented in section 4, while in section 5 we describe how we measure privacy. section 6 provides an

[4] https://www.facebook.com/business/learn/facebook-ads-pixel

extensive analysis of our experiments along with the extracted conclusions. In section 7, we provide the details of the protocol running in the core of `MetaPriv`, which ensures proper and secure communication of users with Facebook through `MetaPriv`. The description of the protocol is coupled with a robust theoretical analysis about security. The paper's final conclusions are presented in section 8.

2 Related work

A large number of research offers users a more private experience on Facebook and other social networks. `FaceCloak` [7] protects user privacy on the SN by shielding personal information from the SN and unauthorized users, while maintaining the usability of the underlying services. `FaceCloak` achieves this through providing fake information to the SN and storing sensitive data in an encrypted form on a separate server. It is implemented as a Firefox browser extension for Facebook. `FaceCloak`'s user privacy attempt resembles our work. However, its main purpose is to hide specific data such as age, name, email address etc. and not user interests derived from interaction with the SN.

`Scramble` [3] allows users to enforce access control over their data. It is a SN-independent Firefox extension allowing users to define access control lists (ACL) of authorised users for each piece of data, based on their preferences. In addition to that, it also allows users to encrypt their posted content in the SN, therefore guaranteeing confidentiality of user data against the SN. The tool allows users to hide information through cryptography. This may require prior knowledge, which is usually counter intuitive for ordinary users.

In [9], the authors test protesting against data labouring [2]: they utilize user interactions with different services as input for training user profiling algorithms. They simulate data strikes against recommendation systems under various conditions. Their results imply that data strikes can put a certain pressure on technology companies and that users have more control over their relationship with said companies. Our work can also be viewed as a protest against the data labouring of users on an SN: if enough users had access to noise attributes, the recommendation systems of Facebook would most likely be disrupted even for new users not using our tool.

Finally, the most relevant work would be `AdNauseam` [5] – a free browser extension designed to obfuscate browsing data and protect user-tracking by advertising networks. It clicks on every displayed ad in different web pages, thereby diminishing the value of all ad clicks – obfuscating the real clicks with clicks that are generated by the tool. Our tool is designed and based on a similar idea.

3 System Model

We will now proceed with introducing the system model we are considering by explicitly describing the main entities participating in the design of `MetaPriv` as well as their capacities.

Social Network (SN): Defined as a graph $\mathcal{G} = (\mathcal{U}, \mathcal{R})$ where the vertices are comprised of users from a set $\mathcal{U}$, with the edges being the relationship between said users, described by the set $\mathcal{R} \subseteq \{\{u, v\} \mid u, v \in \mathcal{U} \text{ and } u \neq v\}$.

Users: Let $\mathcal{U} := \{u_1, \ldots, u_n\}$ be the set of all users registered in an online social network (SN) such as Facebook. Each user has a unique identifier $i \in [1, n]$. In addition to that, each user is associated with a number of attributes. The set of all attributes associated with a user u_i is denoted as $\mathcal{A}_i \subseteq \mathcal{A}$.

Attributes: The set of all available attributes in an SN is denoted by $\mathcal{A} := \{a_1, \ldots, a_m\}$ and is called the attribute space. An attribute is a specific trait that a user u_i possesses, e.g. "u_i likes cats".

BOT: An entity that adds noise to a user profile (u_i). It works by mimicking the user's interaction with the SN and generates noise attributes on their behalf.

User Real and Noise Attributes: Assume a user u_i with a list of attributes $\mathcal{A}_i$. Elements of $\mathcal{A}_i$ may have been generated legitimately (i.e. through the user's real activity) or by the `BOT`. The set of all attributes generated by the user's legitimate activity is denoted as $\mathcal{A}_i^r \subseteq \mathcal{A}_i$ while the set of all attributes associated with u_i but generated by the `BOT` is denoted by $\mathcal{A}_i^n \subseteq \mathcal{A}_i$

3.1 High-Level Overview

The core idea of `MetaPriv` is to fuddle Facebook's opinion about a user u_i by obfuscating u_i's real attributes $\mathcal{A}_i^r$ with the help of noise attributes $\mathcal{A}_i^n$. To that end, we use the `BOT` and have it interact with the SN on behalf of u_i. Ideally, to achieve privacy, the amount of traffic generated by the `BOT` should be the same or more than the traffic generated by u_i.

When user u_i creates an account on Facebook they have no attributes (i.e. the set $\mathcal{A}_i$ is empty). Following registration, u_i begins generating activity (e.g. adding friends, liking pages and posts). By collecting and analyzing user activities, Facebook creates a list of attributes that each user is potentially interested in. a_1 – "u_i likes cooking" These attributes are considered as real and added to the set $\mathcal{A}_i^r$ -a subset of $\mathcal{A}_i$, i.e. $\mathcal{A}_i^r \subseteq \mathcal{A}_i$. The set $\mathcal{A}_i$ is then used by Facebook to decide which posts and advertisements are presented in the respective u_i feed. In this scenario, all u_i's interests are known to the SN, which can make accurate predictions about their preferences and therefore populate their account with accurate personalized content. In this work, we are examining ways of protecting user privacy from a potentially malicious or at least curious SN. To achieve this, we have created `MetaPriv`[5]. With this service users can confuse an SN about their real interests. `MetaPriv` revolves around the following simple idea: Since the SN personalizes users by analyzing their activities on the platform, the sole action required is to generate noise traffic on behalf of a user. This will result in adding attributes to the set $\mathcal{A}_i^n$ containing the noise attributes described earlier.

[5] `MetaPriv` is the name given to the tool and `BOT` is the tool's main functionality -i.e. the part of the tool generating the noise.

With this in mind, we built a BOT in the core of MetaPriv, whose functionality is described below.

First, the BOT needs access to u_i's account. This can be done in two ways: Either through u_i's credentials or through their browser profile folder i.e. the hidden folder in an operating system's user folder, where all web browser cookies, toolbars, extensions etc. are stored. Hence, the BOT's account access input is either the credentials or the path to the profile folder.

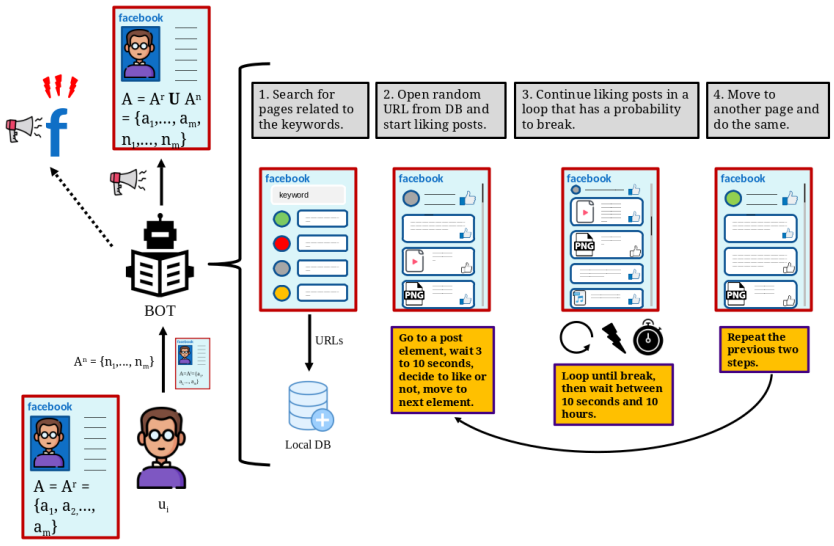


Fig. 1. High-level overview of the BOT's functionality.

After the BOT gains access to the user account, it requires a set of keywords generated by a different part of MetaPriv that serve as noise attributes. The keyword generator, however, requires a seed keyword that the user must input at least once.

The last input is the desired level of privacy by the user. This refers to the level of convenience and benefits that users are willing to lose in order to better protect their privacy. This is further described in section 5. Finally, the BOT repetitively executes nine steps (as shown below). Figure 1 is a graphical representation of the BOT's functionality that depicts the process of adding noise attributes to a user profile.

BOT's Main Steps

Step 1 Opens a browser instance – Firefox or Chrome/Chromium.
Step 2 Opens `https://www.facebook.com` and logs in.
Step 3 Opens `https://www.facebook.com/search/pages?q=keyword`, collects data related to that keyword and stores the corresponding URLs in a database.
Step 4 Selects a random URL from the database and opens it in the browser.
Step 5 Clicks the page's "like" button then moves to the first post element.
Step 6 Waits between 3 and 10 seconds. Decides whether to like the post, then moves to the next element.
Step 7 Repeats step 6 in a loop. When moving to the next element, the page will scroll down, loading more post elements. This is referred to as a scrolling loop.
Step 8 Breaks the loop based on an inputted privacy value, then goes back to step 4 and repeats the remaining steps.

4 Threat Model

We consider an attacker $\mathcal{ADV}$ who can act in the following malicious ways:

1. Corrupt the entire SN and break user privacy by learning their interests;
2. Break user privacy by corrupting `MetaPriv` and finding the noisy attributes created for each user;
3. Change the user's noisy data mainly to gain profit or market advantage against competitors (e.g. change noisy data to present targeted advertisements to users).

Based on these three malicious behaviours, we have defined a set of attacks available to $\mathcal{ADV}$.

Definition 1 (Traditional Profiling Attack). *Let u_i be a legitimate user signed on to a social network platform SN. In addition to that, let $\mathcal{ADV}$ be a curious adversary that has corrupted SN. Then, $\mathcal{ADV}$ can successfully launch a Traditional Profiling Attack, if, for each registered user, u_i gets a detailed and accurate list by SN with the entire activity of u_i in the platform (e.g. likes, follows, posts, etc.).*

Definition 2 (Noise Data Substitution Attack). *Let $\mathcal{ADV}$ be an adversary that overhears the communication between the `BOT` and a user u_i. In addition, assume that u_i wishes to add an attribute a_f to u_i's list $\mathcal{A}_i^n$. $\mathcal{ADV}$ successfully launches a Noise Data Substitution Attack, if they manage to replace the attribute a_n with another of their choice without the `BOT` realizing it and eventually adding it to `MetaPriv`'s database for user u_i.*

Definition 3 (Noisy Attribute Identification Attack). *Let $\mathcal{ADV}$ be a malicious adversary, who overhears the communication between the* `BOT` *and a legitimate user u_i. Additionally, assume that $\mathcal{ADV}$ gains access to the database where u_i's data generated by* `MetaPriv` *is stored. Then, $\mathcal{ADV}$ launches a successful Noisy Attribute Identification Attack either by intercepting the exchanged messages between u_i and the* `BOT` *or by examining stored user data and correctly finding at least some of the noisy attributes used by* `MetaPriv`*.*

5 Measuring User Privacy on Facebook

Previous works focus on measuring privacy according to the visibility and sensitivity of user attributes [1,8,4]. This approach, however, is inapplicable, as the aim is to confuse the data collector, thus leading to inaccurate user profile predictions. Visibility of a user's attributes would always be maximum, since the SN stores all user interactions with it. Additionally, in this work the concept of sensitivity cannot apply, since all user attributes are known to the SN (i.e. can be considered public). With this in mind, we propose a new definition for privacy on an SN based on a user's *real* and *noisy* interactions with the SN. Real interactions are daily, *legitimate* user interactions with the SN. Noisy interactions are `BOT`-produced and mainly generate fake activity on a user's profile.

Our first approach on quantifying privacy was characterized by rather elementary and naive thinking: Initially, we defined the notion of *Theoretical Privacy*. The intuition behind Theoretical Privacy was that a user's level of privacy is proportional to the number of noise in their profile. However, the results of our first experiments did not support this. Apparently, the time that a user likes a post, a page, etc. seems to be significant for Facebook's personalization algorithms. More precisely, it seems that Facebook weighs a user's recent rather than older content. In view of the above, we refined our idea on quantifying privacy and defined *Effective Privacy* -an alternative that better fits Facebook's models.

Definition 4 (Theoretical Privacy). *Theoretical privacy is measured by taking into account the amount of posts liked by a user u_i and the* `BOT`*. User u_i's theoretical privacy with $j + k$ attributes is defined as:*

$$P_i^{th} = \frac{\sum_{j \in \mathcal{A}_i^r} RA_j^{th} - \sum_{k \in \mathcal{A}_i^n} NA_k^{th}}{T}, \tag{1}$$

where RA_{th} is the number of specific attribute-related posts liked by u_i, NA_{th} is the number of specific attribute-related posts liked by the `BOT` *and T is the total number of posts liked by u_i's account.*

Definition 5 (Effective Privacy). *For this definition we consider the effective strength of user real and noise attributes. The strength of a user's real attribute is proportional to:*

- *the number of posts in the main feed from liked pages linked to an attribute. Variable: r_p*

- *the number of recommended/suggested posts in the main feed from pages linked to an attribute, but not liked by the user or the* `BOT`*. Variable:* r_{sp}
- *the number of video posts from the main video feed (https://www.facebook.com/watch) linked to an attribute. Variable:* r_{vp}
- *the number of video posts from the latest video feed (https://www.facebook.com/watch/latest) linked to an attribute. Variable:* r_{lvp}

The effective strength of a real attribute is defined as:

$$RA_{eff} = \frac{1}{n}\left(a\frac{r_p}{t_p} + b\frac{r_{sp}}{t_{sp}} + c\frac{r_{vp}}{t_{vp}} + d\frac{r_{lvp}}{t_{lvp}}\right), \tag{2}$$

where $a, b, c, d \in \{0, 1\}, n = a+b+c+d$, t_p *is the total number of posts shown in the main feed,* t_{sp} *is the total number of suggested posts shown in the main feed,* t_{vp} *is the total number of video posts related to* u_i*'s attributes from the main video feed and* t_{lvp} *is the total number of video posts from the latest video feed. Each of the variables* a, b, c, d *is given the value 0, when their respective fraction is 0. Otherwise they are given the value 1. This is done so that, if one effective strength variable has a value of 0 (i.e. no posts), then it will not be taken into account for the final effective privacy value.*

A similar definition stands for the effective strength of noise attributes NA_{eff}*. variables* r_p, r_{sp}, r_{vp} *and* r_{lvp} *are replaced with corresponding noise attributes i.e.* n_p, n_{sp}, n_{vp} *and* n_{lvp}*. The strength of a noise attribute is defined as:*

$$NA_{eff} = \frac{1}{n}\left(a\frac{n_p}{t_p} + b\frac{n_{sp}}{t_{sp}} + c\frac{n_{vp}}{t_{vp}} + d\frac{n_{lvp}}{t_{lvp}}\right) \tag{3}$$

Finally, for a user u_i *with* $j+k$ *attributes, we combine the two variables and reach the effective privacy:*

$$P_i^{eff} = \sum_{j \in \mathcal{A}_i^r} RA_j^{eff} - \sum_{k \in \mathcal{A}_i^n} NA_k^{eff} \tag{4}$$

In both cases, the resulting value will be $P \in [-1, 1]$. The closer it is to 0, the more indistinguishable will the noise attributes be from real attributes. Therefore, the account of an arbitrary user u_i is private *iff* $P \approx 0$ or $P \leq 0$.

6 Implementation and Results

To demonstrate `MetaPriv`'s functionality, we created a new Facebook account and ran a 10-week experiment, building the account's real and noise attributes. Additionally, we tried different scenarios to test efficiency and estimate the time required for users to retrieve their privacy.

We used `MetaPriv` to simulate both user and `BOT` interactions[6] with Facebook. The program was implemented using Python 3.10 and Selenium WebDriver -a framework for testing web applications. Selenium provides libraries in

[6] We now make a clearer distinction between `MetaPriv` and the `BOT` as `BOT` interactions are now used to refer to the noise traffic generated by `MetaPriv`.

several languages (e.g. python, Java, C++) allowing simulation of an automated user interaction with a webdriver -a web browser that can be controlled remotely (e.g. geckodriver, chromedriver etc.).

Open Science and Reproducible Research: Our source code[7] is available to support open science and reproducible research. Interested reviewers can also download our application for possible testing or a simple overview of the generated research artifact.

6.1 Experiments and Attribute Strength Results

The Facebook user we created for our experiments is a 22-year-old female from Ireland (the account and all interactions were made through an Azure server with an Irish IP address). At the end of each week, we ran an analysis of Facebook's main, video and latest video feed by opening the respective URLs, going through a certain amount of posts in them and saving information about said posts in an SQL database. The information saved would be: *(1)* page URL, *(2)* post URL, *(3)* time of publication, *(4)* text from the post and *(5)* screenshot of the post.

Weeks 1 & 2: The first two weeks consisted of building the profile with just one attribute. More precisely, we used the attribute "cat" to have Facebook associate our user with cats. We provided the keyword "cat pictures" as input to `MetaPriv`. The program liked 1,056 posts from 51 keyword-related pages over two weeks. This keyword would serve as the user's *real attribute*. After one week, 'Recommended' posts appeared in the main feed. Out of 264 posts, 32 were recommended and 11 seemed relevant to the user's profile:

> 1 post related to demographics -a house in Dublin; 1 post about cats from a page about cats; 2 posts about tigers (both from Facebook group: WildCat Ridge Sanctuary); 1 post about demographics and cats (page name: North Dublin Cat Rescue Ireland); 1 post about ostriches, 1 about bulls, 2 about dogs, 1 about rare animals (related to animals); 1 post about "Dads Acting Like Their Teenage Daughters" (possibly gender-related).

Other recommended posts were unrelated to "cats" and had a dozen million views (thus they were most likely trending). Almost all recommended posts were videos.

Two weeks later, we analyzed 449 posts from the main feed and got 13 recommended posts along with 23 "join group" recommendations from cat-related Facebook groups. 8 of the recommended posts were linked to the user's profile:

> 1 post related to demographics: Football game GERMANY vs IRELAND (2002); 1 post about cats from Facebook group: CAT LOVERS PHILIPPINES; 4 posts about animals from a group about animal comics; 1 post about cats from the 'Daily Mail' page; 1 post from a group about Dinosaurs. The name of the person posting was: Margaret Happycat.

[7] https://github.com/ctrgrb/MetaPriv

This time, most recommendations appeared from groups, though the user was not a member of any.

Week 3: We added a second keyword as a noise attribute to the profile. At this point, the noise was manifested through liking a noise-related page and its posts at every 10th page switch. In essence, 10% of the interactions with Facebook were now related to one noise attribute. This 10% represented 72 out of 554 posts liked on week 3 from 5 pages linked to the keyword "guns"[8]. This time, there were no recommended posts. An analysis of 547 posts from the main feed showed that 19 were linked to the noise attribute. The latest video feed contained only 21 videos from liked pages related to the real attribute (i.e. cats). In the main video feed, we analyzed 184 video posts. 70 of them included words such as: ['cat','Cat','kitten','Kitten'] in their description or page URL and were, thus, related to the real attribute, while nothing was related to the noise attribute.

Week 4: We increased the noise amount to 20%. From 530 liked posts, 112 came from 8 pages related to the noise attribute. In the main feed, from 337 posts, 38 were from pages related to the noise attribute. Facebook stopped showing recommended posts at this point, however, 'Suggested for you' posts began to show. Out of the 337 posts, 8 were labeled as 'Suggested' out of which 1 was related to animals, 3 specifically to cats and the remaining were possibly gender-related. This time too, the latest video feed showed only cat-related videos and in the main video feed, out of 152 videos, 35 included the words: ['cat','Cat','kitten','Kitten'] in the description or page URL, while no videos were related to guns.

Week 5: We decided to add another noise attribute, thus dividing Facebook interaction as follows: 70% cats, 20% guns and 10% cooking. From a total of 485 liked posts, 130 were related to the keyword "guns" and 36 to "cooking recipes". This time, out of 673 posts in the main feed, 67 were related to guns and 147 to cooking. Our theory for increased cooking content is that a cat lover is more likely to also like cooking rather than guns.[9]. This time, out of 16 suggested posts, 14 were cats. In the latest video feed, out of 51 videos, 21 were cats, 1 guns and 26 cooking. Finally, in the main video feed, out of 136 posts, 27 were cats, 3 guns and 7 cooking.

Week 6: We increased the amount of noise for the cooking attribute to 20% and the gun attribute to 30%, thus dividing Facebook interaction as follows: 50% cats, 30% guns and 20% cooking. From a total of 647 liked posts, 213 were guns and 125 cooking. In the main feed, out of 405 posts, 35 were guns and 66 cooking. There were also 7 suggested posts, out of which 4 were cooking and 2 cats. In the latest video feed, out of 65 posts, 12 were cats, 2 guns and 51 cooking. Finally, in the main video feed's 103 posts, 27 were cats and 15 cooking.

[8] It is worth noting that the percentage value is an approximation since `MetaPriv` is designed with randomness in mind to avoid patterns in its behaviour.

[9] This might also be related to the fact that Ireland has one of Europe's least permissive firearm legislation – hence gun-related content is heavily regulated.

Week 7: We added another noise attribute that would be stronger than others. Hence, Facebook interaction became: 23% cats, 23% guns, 23% cooking and 30% chess. From a total of 365 liked posts, 90 were cats, 89 guns, 76 cooking and 110 chess. The main feed's 286 posts were divided as follows: 45 guns, 72 cooking and 2 chess. From 14 suggested posts, 10 were cooking and 1 chess. In the latest video feed, out of 162 posts, 18 were cats, 35 guns, 83 cooking and 22 chess. The 137 posts in the video feed were divided as follows: 25 cats, 1 guns, 9 cooking and 1 chess.

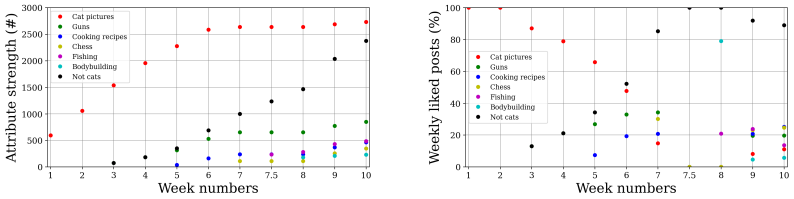
Week 8: The aim was to examine results, when new attributes were added without reinforcing old ones. For the first half of the week Facebook interaction was 100% fishing-related and the second half 20% fishing and 80% bodybuilding.

- First half: Liked 235 posts about fishing. In the main feed, out of 402 posts, 207 were cats, 45 guns, 115 cooking, 4 chess and 15 fishing. Out of 7 suggested posts, 4 had to do with fishing and the others were unrelated to the user's attributes. In the latest video feed, from 190 videos, 14 were cats, 48 guns, 72 cooking, 39 chess and 18 fishing. In the main video feed, out of 148 videos, 12 were cats, 2 guns, 10 cooking, 3 chess and 1 fishing.
- Second half: Liked 48 fishing posts and 181 bodybuilding posts. In the main feed, out of 423 posts, 229 were cats, 33 guns, 127 cooking, 22 fishing and 7 bodybuilding. Out of 2 suggested posts, 1 was bodybuilding and the other unrelated. In the latest video feed, out of 156 videos, 16 were cats, 9 guns, 30 cooking, 34 fishing and 72 bodybuilding. In the main video feed, out of 128 videos, 1 was cats, 2 guns, 20 cooking, 1 chess and 1 fishing.

Week 9: We ran `MetaPriv` with 10% cat-related traffic and the remaining with the following noise attribute layout: 20% guns, 20% cooking, 20% chess, 20% fishing, 10% bodybuilding. From 626 liked posts, 51 were about cats, 122 guns, 130 cooking, 144 chess, 149 fishing and 29 bodybuilding. In the main feed, out of 460 posts, 199 were about cats, 51 guns, 145 cooking, 19 chess, 25 fishing and 7 bodybuilding. This time there were no suggested posts. In the latest video feed, from 154 videos, 18 had to do with cats, 14 guns, 77 cooking, 35 chess and 18 fishing. In the main video feed, from 137 videos, 25 were about cats, 1 guns, 9 cooking and 1 chess.

Week 10: In the last week we ran `MetaPriv` with the same parameters as in week 9: 10% cats, 20% guns, 20% cooking, 20% chess, 20% fishing and 10% bodybuilding. From 381 liked posts, 42 were cats, 75 guns, 96 cooking, 94 chess, 52 fishing and 22 bodybuilding. In the main feed, out of 442 posts, 160 were cats, 71 guns, 139 cooking, 30 chess, 32 fishing and 4 bodybuilding. Again, there were no suggested posts. In the latest video feed, from 133 videos, 10 were cats, 15 guns, 75 cooking, 22 chess and 12 fishing. Finally, in the main video feed, from 124 videos, 6 were cooking, 1 chess and 2 bodybuilding.

The total amount of posts liked on a weekly basis for each attribute (attribute strength), is shown in Figure 2b. The week number is noted on the horizontal axis and the attribute strength (total amount of posts liked) on the vertical axis. As the figure indicates, even on week 10, the "cat' attribute strength outweighs



(a) Weekly progression of theoretical attribute strength

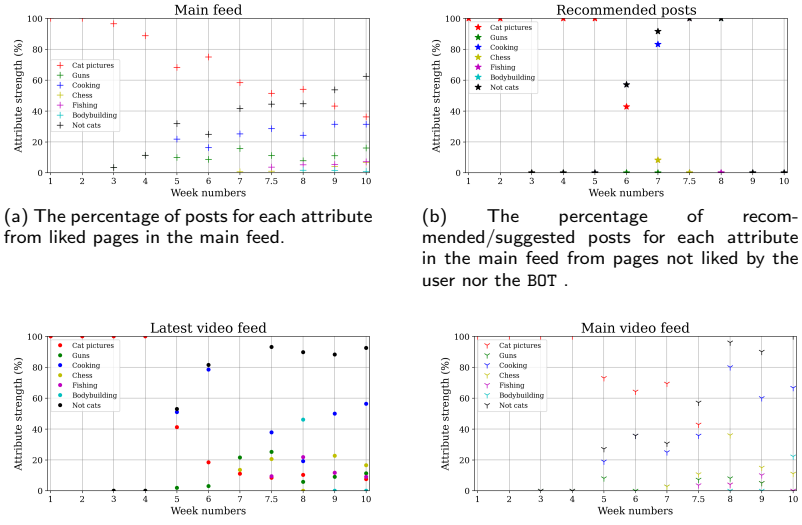
(b) Ratio of weekly liked posts

Fig. 2. The total amount of posts liked and the ratio of posts liked per week.

all others combined, since the attribute remained reinforced even when said reinforcement decreased over time.

Figure 2b represents the ratio of posts. Here, the ratio is calculated using the posts liked on a specific week, omitting those of previous weeks. This time, the attribute strength on the vertical axis stands for the percentage of liked posts for each attribute.

Next, we present the results of each variable for the effective attribute strength. The main feed, recommended posts, latest video feed and main video feed are represented in Figure 3 along with the combined noise attribute strength.



(a) The percentage of posts for each attribute from liked pages in the main feed.

(b) The percentage of recommended/suggested posts for each attribute in the main feed from pages not liked by the user nor the BOT .

(c) The percentage of video posts for each attribute from the latest video feed.

(d) The percentage of each attribute of video posts from the main video feed.

Fig. 3. Effective attribute strength variables with combined noise

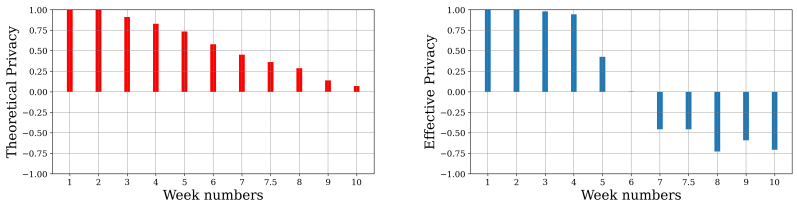
We can, now, compare results between Figure 3 and Figure 2b: on weeks 5 to 8, noise-effective attribute strength variables approached real variables. Figure 2b shows that around week 6, there are more noise-related likes than real likes. Consequently, Facebook's recommendations show more noise-related content as we can see from Figure 3. In the first 4 weeks, Figures Figure 3c and Figure 3d show no relation to noise attributes. We thus conclude that 20% noise is not enough to change said variables. Also, Figure 3b shows that in a few weeks' time, there were no recommended/suggested posts in the main feed (weeks 3, 9 and 10).

To avoid confusion in Figure 3 we must clarify that in the main video feed Figure 3d and the recommended/suggested posts Figure 3b, the Facebook content is derived from pages not liked by the user. The content is both user attribute-related and unrelated. It is assumed that the unrelated content is presented by Facebook because of other features in their recommendation systems e.g. users who liked X also liked Y. Their recommendation algorithms are not open source, hence their mode of operation is concealed. Due to this, our results are based on content exclusively related to user attributes.

6.2 Privacy Results

Based on the definitions described in section 5, we have calculated each week's Theoretical (Figure 4a) and Effective Privacy (Figure 4b) values.

On the first two weeks we built the user's real attributes and added increasing noise to render Facebook's noise feed equal to the real.



(a) Theoretical privacy (b) Effective privacy

Fig. 4. Theoretical and Effective privacy results

As expected in section 5, Effective Privacy in week 6 (50% noise) is close to 0. Once the amount of real traffic generated by users equals the amount of noise traffic, users achieve privacy. The theoretical real attribute strength outweighs the combined noise attribute strengths even after 10 weeks, as shown in Figure 2a. This explains the difference between the Theoretical and Effective Privacy values and shows that Facebook emphasises on the user's recent interests, suggesting a "time of like" variable in its recommendation systems. This also proves that the Effective Privacy is a more accurate way of measuring privacy on a SN.

We added more noise on week 7 and saw a small decrease in the Effective Privacy value -i.e. the account became more private. On week 8, we stopped reinforcing the real attribute to simulate what would happen if the user took a break from Facebook, while the BOT ran. We noted significant decrease in the Effective Privacy value.

Finally, on weeks 9 and 10, we simulated a rarely active user combined with BOT background activity (90% noise). On week 9, the Effective Privacy value increased as the real attribute was reinforced again. The Effective Privacy value decreased again on week 10.

7 Protocol and Security Analysis

In this section, we formalize the communication between a user u_i and BOT by describing the protocol running the core of MetaPriv. Furthermore, we prove our construction's security against the threat model defined in Section 3.

7.1 Protocol

We assume the existence of an IND-CPA secure symmetric key encryption scheme $\mathsf{SKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. Moreover, we further assume that u_i and the BOT communicate over a symmetrically encrypted channel, using a shared symmetric key $\mathsf{K}_{\mathsf{u_iB}}$ generated as $\mathsf{K}_{\mathsf{u_iB}} \leftarrow \mathsf{SKE.Gen}(1^\lambda)$, where λ is the security parameter of SKE.

The protocol is initiated by u_i each time they add a new attribute to the BOT's list $\mathcal{A}_i^n$. To do so, u_i picks an attribute $\mathtt{att}_i$, encrypts it using $\mathsf{K}_{\mathsf{u_iB}}$ as $c_{\mathtt{att}_i} \leftarrow \mathsf{SKE.Enc}(\mathsf{K}_{\mathsf{u_iB}}, \mathtt{att}_i)$ and sends the following message to the BOT:

$$m = \langle t, c_{\mathtt{att}_i}, \mathsf{HMAC}(\mathsf{K}_{u_iB}, t \| c_{\mathtt{att}_i}) \rangle,$$

where t is a timestamp and HMAC is a keyed-hash message authentication code operating as a pseudorandom function (PRF). Upon receiving m, the BOT verifies the freshness and integrity of the message by checking the timestamp and the HMAC respectively. If any verification fails, the BOT outputs $\perp$ and aborts the protocol. Otherwise, it stores $c_{\mathtt{att}_i}$ to its list of attributes.

7.2 Security Analysis

Here, we prove the security of our construction against the threat model of Section 3. We begin this Section with a brief discussion on the *Traditional Profiling Attack* as per Definition 1.

Traditional Profiling Attack: To successfully perform a *Traditional Profiling Attack* against a user u_i, an adversary $\mathcal{ADV}$ needs a detailed list by the SN containing u_i's full activities. However, our extensive experimental control shows that our construction can achieve full privacy after 6 weeks. As discussed

in Section 6.2, *effective privacy* is a more accurate index for Social Networks compared to *theoretical privacy*. Hence, we can conclude that a user u_i can fully prevent a *Traditional Profiling Attack* through our construction after 6 weeks. However, since our construction allows users to ***quantify*** their privacy, each user can prevent the attack fully or partially or refrain from preventing it.

Proposition 1 (Noise Data Substitution Attack Soundness). *Let $\mathcal{ADV}$ be a malicious adversary overhearing communication between a user u_i and the* BOT*. Moreover, let* SKE *be an IND-CPA secure symmetric-key cryptosystem and HMAC a key-message authentication code, proved to be a PRF. Then $\mathcal{ADV}$ cannot successfully perform a Noise Data Substitution Attack.*

Proof. $\mathcal{ADV}$ will successfully launch a Noise Data Substitution Attack if they tamper with message $m = \langle t, c_{\mathtt{att}_i}, \mathsf{HMAC}(\mathsf{K}_{u_i B}, t \| c_{\mathtt{att}_i}) \rangle$]sent by u_i to the BOT. To do so, $\mathcal{ADV}$ must satisfy at least one of the following:

C1: Replace $c_{\mathtt{att}_i}$ with another ciphertext $c_{\mathcal{ADV}}$ encrypting an attribute of their choice;
C1: Replay an old message.

- **C1** will hold, if $\mathcal{ADV}$ *(1)* picks an attribute $\mathtt{att}_{\mathcal{ADV}}$ of choice, *(2)* gets the symmetric key copy $\mathsf{K}_{\mathsf{u_i,B}}$, *(3)* encrypts $\mathtt{att}_{\mathcal{ADV}}$ using $\mathsf{K}_{\mathsf{u_i,B}}$, *(4)* generates a valid HMAC and *(5)* swaps message components m with malicious ones. However, given the IND-CPA security of SKE, $\mathcal{ADV}$ can only recover the symmetric key with probability negligible in λ, where λ is the security parameter of SKE. Thus, $\mathcal{ADV}$ can satisfy **C1** only with negligible probability.
- The other option for $\mathcal{ADV}$ is to replay an older valid message: $\mathcal{ADV}$ intercepts m and replaces it with a previously intercepted m_{old}. However, since the HMAC portion of the message contains a timestamp, $\mathcal{ADV}$ would need to create a new valid HMAC with a new timestamp. Similarly to **C1**, this can only occur with knowledge of $\mathsf{K}_{\mathsf{u_i,B}}$ and hence, with negligible probability.

As a result, both **C1** and **C2** can be satisfied with negligible probability, and thus, $\mathcal{ADV}$ can launch a successful Noise Data Substitution Attack only with negligible probability. □

Proposition 2 (Noisy Attribute Identification Attack Soundness). *Let $\mathcal{ADV}$ be a malicious adversary overhearing communication between u_i and the* BOT *and having access to the* BOT*'s database. Let* SKE *be an IND-CPA secure symmetric-key cryptosystem. Then $\mathcal{ADV}$ cannot launch a successful Noisy Attribute Identification Attack.*

Proof. Attributes are both transferred, stored and encrypted under $\mathsf{K}_{\mathsf{u_i,B}}$. Hence, even if $\mathcal{ADV}$ intercepts the message m sent from u_i to the BOT, access to $\mathsf{K}_{\mathsf{u_i,B}}$ is required to recover the attribute's value. Similarly, even with access to the BOT's database, $\mathcal{ADV}$ would still need $\mathsf{K}_{\mathsf{u_i,B}}$ to decrypt all stored ciphertexts. However, given the IND-CPA security of SKE, $\mathcal{ADV}$ can only recover $\mathsf{K}_{\mathsf{u_i,B}}$ with probability negligible in λ. Hence, $\mathcal{ADV}$ can launch a successful Noisy Attribute Identification Attack only with negligible probability. □

8 Conclusion and Societal Impact

Social networks shaped the digital world becoming an indispensable part of our daily lives. Over the years, these platforms have gained a reputation for tracking user online activity. These strategies may prove threatening for multiple spheres of peoples' lives – spanning from consumption to opinion formation – and may have ominous effects on democracy [6]. This vast collection of personal data by SNs is often exposed (i.e. sold) to third-party companies.

In addition, SN users do not usually have a say on the information they access, as SNs prioritize the content presented on feeds, based on what users most probably want to see. In other words, SN algorithms seemingly hide content and have a great impact on the information users are able to reach. With privacy and societal concerns over SNs rapidly rising, these platforms are seen as rather controversial.

Having identified these issues, we built `MetaPriv`, a tool that adds new privacy safeguards for SN users aimed at hampering SN ability to serve targeted content. `MetaPriv` allows users to define their desired level of privacy. In this way `MetaPriv` strikes a balance between privacy and functionality. We believe this feature will be used in several services in the near future and will help towards building less biased SNs, while minimizing the amount of personal information processed by platforms.

References

1. Aghasian, E., Garg, S., Montgomery, J.: User's privacy in recommendation systems applying online social network data, a survey and taxonomy (2018) 7
2. Arrieta-Ibarra, I., Goff, L., Jiménez-Hernández, D., Lanier, J., Weyl, E.G.: Should we treat data as labor? moving beyond "free". AEA Papers and Proceedings (2018) 3
3. Beato, F., Kohlweiss, M., Wouters, K.: Scramble! your social network data. In: Fischer-Hübner, S., Hopper, N. (eds.) Privacy Enhancing Technologies. pp. 211–225. Springer Berlin Heidelberg, Berlin, Heidelberg (2011) 3
4. Domingo-Ferrer, J.: Rational privacy disclosure in social networks. In: International conference on modeling decisions for artificial intelligence. Springer (2010) 7
5. Howe, D.C., Nissenbaum, H.: Engineering privacy and protest: A case study of adnauseam. CEUR Workshop Proceedings **1873**, 57–64 (2017) 3
6. Khan, T., Michalas, A., Akhunzada, A.: Fake news outbreak 2021: Can we stop the viral spread? Journal of Network and Computer Applications **190**, 103112 (2021) 16
7. Luo, W., Xie, Q., Hengartner, U.: Facecloak: An architecture for user privacy on social networking sites. In: In Proceedings of 2009 IEEE International Conference on Privacy, Security, Risk and Trust (PASSAT-09). p. 1 (2009) 3
8. Maximilien, E.M., Grandison, T., Liu, K., Sun, T., Richardson, D., Guo, S.: Enabling privacy as a fundamental construct for social networks. In: 2009 International Conference on Computational Science and Engineering. vol. 4. IEEE (2009) 7
9. Vincent, N., Hecht, B., Sen, S.: "data strikes": Evaluating the effectiveness of a new form of collective action against technology companies. In: The World Wide Web Conference. p. 1931–1943. WWW'19, ACM, New York, NY, USA (2019) 3