# Tinder Me Softly – How Safe Are You *Really* on Tinder?

Mark Carman[1] and Kim-Kwang Raymond Choo[1,2(✉)]

[1] School of Information Technology and Mathematical Sciences, University of South Australia, Adelaide, Australia
carmd006@mymail.unisa.edu.au, raymond.choo@fulbrightmail.org
[2] Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, USA

**Abstract.** There are known privacy concerns with the use of Tinder, a popular dating app. In this paper, we examine previous attacks on Tinder that have not been documented academically. We also documented the Tinder network API in order to test the previous attacks in a live environment. Although our testing revealed accurate user location data, which was the crux of the prior attacks, has since been patched; we were able to: associate a Facebook profile with a Tinder account due to their shared information, see Facebook pages a user had liked or was a member of, as well as gather user images, which Tinder sends via plain HTTP, for a reverse image search. We also demonstrated the potential for a less accurate location attack that takes into account Tinder's updated security.

**Keywords:** Tinder privacy risks · Dating app risks · Geo-social privacy risks · Mobile app security

## 1 Introduction

Tinder is a mobile application (i.e. an "app") that provides a matchmaking service, which has facilitated over 10 billion matches [1]. Tinder's defining quirk is that matchmaking only occurs between users that are geographically close, and messaging is only allowed between users that encounter and "like" each other through the matchmaking process, which also provides only a single chance to like or pass on each match. Once matched, users are able to freely message each other to chat or to arrange real-world meetings for friendship, dating, or, as identified in popular culture - for casual relationships called "hook-ups" [2].

Privacy has always been a concern with geo-social apps [3, 4]. Tinder and many other dating apps allow users to vet each other before agreeing to meet. However, because Tinder is also a geo-social app, it has an inherent privacy flaw: geo-social apps need to collect location information about their users in order to function. By examining the output of the app, as well as manipulating the input, it is possible that the app might leak personal information about its users. As a trivial example, if you move away from another user until the app no longer considers them nearby, you now know the other user is outside the cut-off distance of the app.

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2017
R. Deng et al. (Eds.): SecureComm 2016, LNICST 198, pp. 271–286, 2017.
DOI: 10.1007/978-3-319-59608-2_15

While rigorous academic research on Tinder has been limited, Tinder is certainly in the public consciousness. The latter is unsurprising, as Tinder attracts thousands of news and blog posts, millions of users, and billions of swipes. While many stories deal with the social aspects and implications of Tinder and related apps, such as discussions on hook-up culture, there are significantly fewer that deal with privacy and security issues surrounding Tinder. Of these, the most crucial involved attacks that were able to obtain the GPS data and Facebook ID of a Tinder user. These attacks provided the motivation for our research. While Tinder has since reportedly patched these vulnerabilities, there is still a question about whether fundamental geo-location information could be exploited to recreate the effects of some of the attacks.

In this paper, we examine and document the network protocol of Tinder to determine what sort of data travels between the Tinder app and the Tinder servers. We then examined how this data might be manipulated to affect the privacy and security of Tinder users. We also conducted tests using Tinder data and our proposed methods in a live environment to get a glimpse at possible real world implications.

## 2 Background and Related Work

### 2.1 Background

Tinder is a free mobile app for both Android and iOS devices, with 50–100 million installs on Android devices alone, at the time of this research [5]. Tinder requires a Facebook account, and uses Facebook permissions when a user creates an account to give Tinder access to their list of friends, photos, and biographical information (e.g. name, age, interests, education, and employment).

Once connected through Facebook, Tinder allows a user to customise their profile from the Facebook entries collected. This includes setting the primary image displayed during matchmaking, as well as up to 5 additional images for their profile. Employment and current academic education are also able to be selected. First name and age are also included on a user's Tinder profile, but are only editable through their Facebook account. In addition to Facebook-sourced information, Tinder allows a user to select their gender, enter a 500 character biography, and connect to an Instagram account.

The ephemeral nature of Tinder's matchmaking comes from the "stack" of potential matches displayed to a user on the main Tinder screen. This stack is a set of recommendations based on a user's age, gender, and distance preferences, as well as Tinder's matchmaking algorithm which includes various hidden parameters, such as "desirability" [6]. For each match in the stack, the user is able to either "pass", "like", or "super like" the prospective match by swiping left, right, and up, respectively.

The profiles themselves are displayed as a stack of cards. While there are superficial differences between the Android and iOS versions, the content of the profiles is the same for both versions. Initially the primary profile picture, first name, and age are displayed, along with buttons to pass, like, super like, and "rewind". Rewind is a feature only available for paying Tinder users, and allows the user to undo a swipe action made immediately prior. Tapping on the profile picture brings up additional information – this includes employment or education information, city, distance between users, photos from a connected Instagram account, if applicable, as well as scrollable slides of up to 5

other images the user has uploaded. Any shared interests, which are Facebook pages both users have liked, as well as shared Facebook friends, are also displayed (Fig. 1).

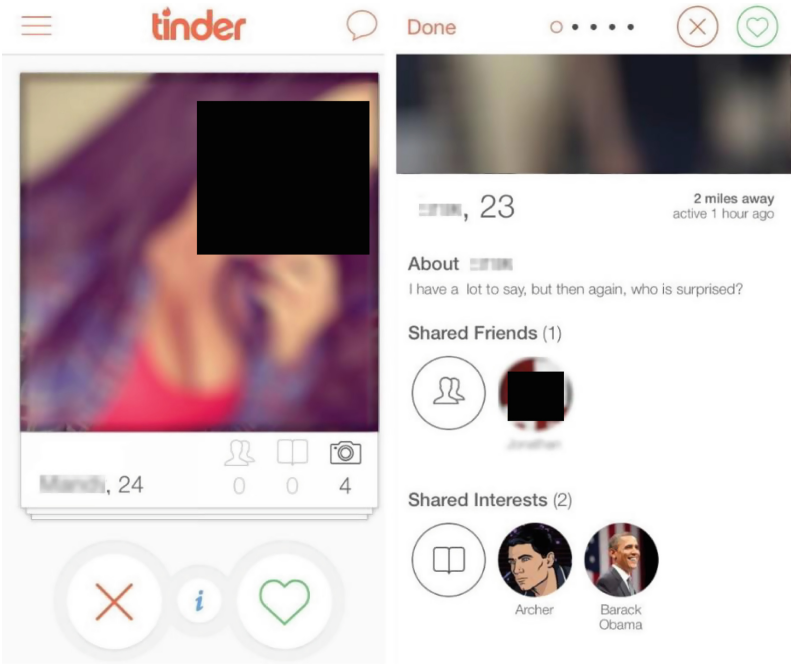


**Fig. 1.** A de-identified example of a Tinder user card. Initial view on left, truncated detailed view on right (Source: http://orzzzz.com/find-out-what-tinder-really-is-before-you-rush-to-join-it.html; Last accessed: 20/07/2016)

Passing or liking a profile removes them from the stack and the next card in the stack is displayed. Once a match has been removed from the stack, it is ostensibly gone forever. However, circumstances such as creating a new account would reset matches.

Liking a profile initially does nothing. However, if the like is reciprocated at any point, Tinder displays a match notification. This then enables messaging between the two users. This is the only time messaging is enabled between users. Once messaging is enabled, users are able to freely message each other through the Tinder app, enabling the ability to continue the conversation, move to a different medium, or setup an off-line meeting for friendship, dating, or a hook-up. Messaging is predominantly text-based, although a partnership with the company GIPHY in 2016 means users can now search and send preselected, "expressive", video images [7]. Either party is able to un-match at any stage, which revokes the ability to send messages (Fig. 2).

A super like is functionally similar to a like, but displays a blue background, a short message, and a star-icon to your potential match when they view your card in their stack. This may seem trivial at first glance, but with messaging disabled between unmatched users, a super like is the only way to signal that you have seen and enjoyed another user's profile. Super likes are limited to 1 per day [8] (Fig. 3).

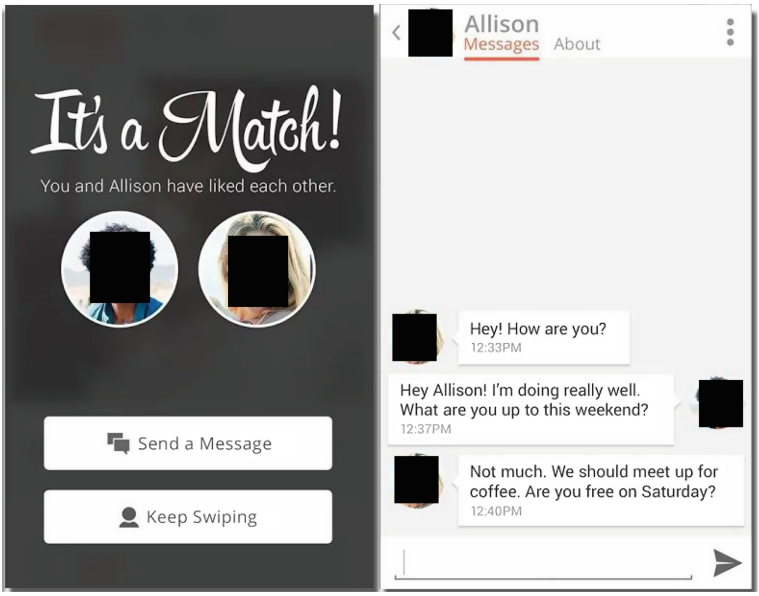


**Fig. 2.** An example Match taking place (Source: Google App Store Tinder page, https://play.google.com/store/apps/details?id=com.tinder; Last accessed 05/07/2016)

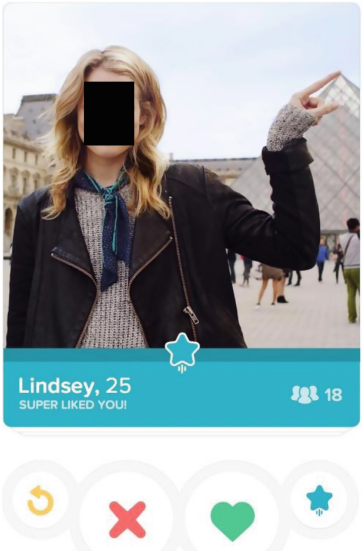


**Fig. 3.** An example of a super like indication. (Source: http://blog.gotinder.com/updated-introducing-super-like-a-new-type-of-swipe/; Last accessed 23/07/2016)

### 2.2 Previous Work

As stated earlier, existing technical research on Tinder is lacking. For example, in the earliest and most dangerous exploit (described more fully below), the only information we could find was obtained from social commentary websites and social media, due to Tinder's infancy at the time. Eventually we came across a news article displaying emails sent to Tinder by the creator of the attack, disclosing its details [9]. Tinder eventually claimed they had fixed these vulnerabilities, but only after a week without reply lead the attacker to go public with his story [9]. Because the vulnerability was only publicly known for a day, and the only evidence of it consisted of emails to and from Tinder (rather than a well-documented formal disclosure), there was no way to verify the correctness of the claims made. However, given that the server no longer returned the location data when we tested it - see Fig. 4, and the fact that details of the attack were reported on several social news websites, this attack appears to be authentic.

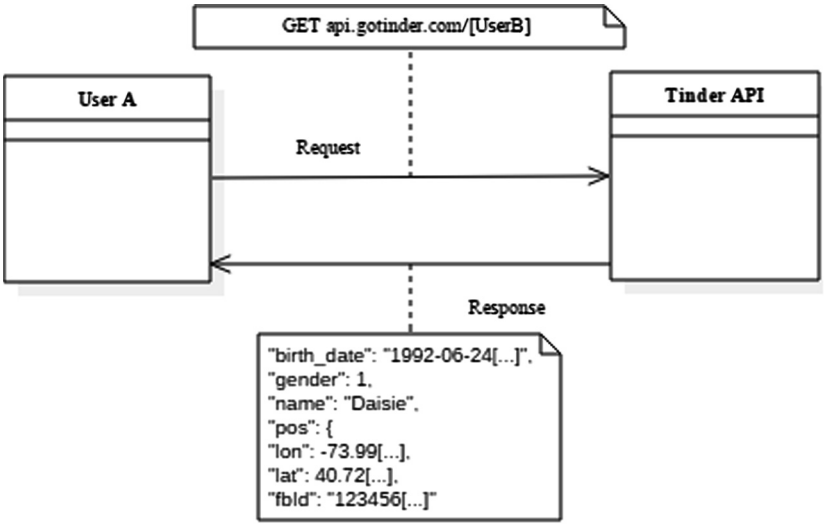


**Fig. 4.** The first major Tinder attack. GPS and Facebook ID data were returned when getting a user profile

The attack as described was completely straightforward. When the Tinder Application Programming Interface (API) requested a user profile (which the app does to display its profile stack), the API returned extra information. Unfortunately, this extra information was in the form of exact GPS co-ordinates, and a Facebook ID, which the attacker was able to read by examining the network traffic. GPS co-ordinates could obviously then be used to compromise the location of the user. Furthermore, a Facebook ID allowed the attacker to obtain the Facebook profile of the user, which generally contains more personal information than Tinder displays, which may escalate the attack in other ways.

Several months after the attack described above was fixed, a second attack, using a more novel approach was described. While exact GPS co-ordinates were no longer available, the response from the Tinder API included a very accurate distance between two users [10]. The author of this attack utilised this by creating three fake profiles and trilateration to determine the exact location of the user (see Fig. 5). Trilateration is essentially the projection of three circles with radii set to the respective distances reported by Tinder, and then seeing the point where all the circles intersect. Again, the write-up was posted on a security blog (rather than a formal publication), and included a tool, explanation, and video on how to locate your own Tinder account [10].

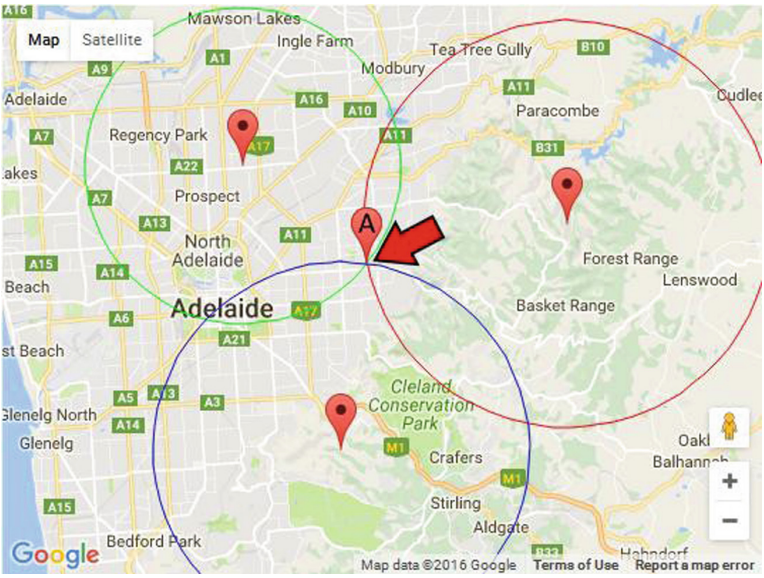


**Fig. 5.** Visualising an example trilateration attack. Point 'A' (University of South Australia) is the point where all three circles intersect

Such vulnerability motivated this research. Tinder also fixed this vulnerability and now only provides approximate numbers in its API. However, we question whether this is sufficient to keeps users safe.

While there have been two major security concerns for Tinder, there have also been a considerable amount of effort invested into gaming the Tinder system. From basic bots that automatically swipe right to give a "like" on every profile to maximise the chance of getting a match [11, 12], to a much smarter bot that learns what facial features you find attractive based on your swipe history, and can even automate conversation openers [13]. Other bots have a financial rather than romantic or sexual interest at heart, where they pretend to be a highly attractive user that sends spam messages to any matches it can make [14]. While Tinder has not implemented any changes that directly target bots, a March 2015 update [15] limited the number of

swipes free users had per day, which significantly reduced the effectiveness of these mass swiping bots. Paid users retained unlimited swipes.

The aim of this study is to determine if there are still privacy concerns when using Tinder.

Specifically, we investigated whether Tinder now prevents the type of attacks previously conducted, and whether other, more subtle attacks, were possible - particularly associated with the following: GPS location, personal user data and social media connections. In order to do this, we used tools to understand how the Tinder API worked and then used that information to examine the quality of Tinder's security, and to propose attacks.

## 3 Experiment Design

A client device (Table 1) was connected to Tinder to document its API. To accomplish this, all of the network traffic from the client device was captured using a proxy, and then documented. Once the API was well understood, we then replicated the phone's functionality by developing a specialised Tinder bot. The areas of the API we focussed on understanding and attempting to exploit were:

- GPS location of users
- Personal user data (Birthday, name, pictures, etc.)
- Social media connections

**Table 1.** Client device details

| Make | Model | OS | Tinder version |
|---|---|---|---|
| Motorola | Moto E | Android 4.4.4 | Tinder 4.5.2 |

### 3.1 Server Set-up

To properly capture the traffic from the Tinder application, a server was required to analyse the network traffic from our client device. We chose a pre-packaged software tool called Charles for this task because it was available for many different platforms, was easy to use, and was able to analyse HTTPS traffic from the client. Because Tinder uses HTTPS, which is encrypted, we needed to use Charles as a man-in-the-middle proxy between the client phone and the Tinder API server (see Fig. 6; similar to the approach in [16]). This set-up allows the proxy to pretend to be the server to the client and the client to the server, allowing decryption for both. Further, we used the same server machine to run our eventual bot, which was developed in C#.

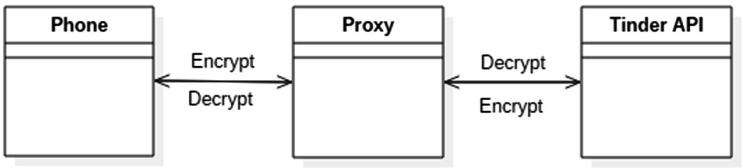


**Fig. 6.** Proxy setup with proxy in between the client device and the Tinder server

### 3.2 Client Set-up

To set-up our client, we first installed Tinder 4.5.2 on the client device. Next, we set our Charles server as the phone's internet proxy using the Wi-Fi settings page in Android, which allows us to examine HTTP traffic on the Charles server. To allow us to examine SSL traffic, we needed to generate a root Charles Certificate Authority Certificate. A Certificate Authority is a trusted authority able to sign and distribute SSL certificates, which is required to decrypt and re-encrypt the traffic from our phone to forward to the Tinder server. After generating this certificate on the Charles computer, we added the certificate as a Certificate Authority through the Android security settings. With both the client and the server correctly configured, we then used Tinder in a normal fashion on the phone for a few minutes while recording all the network traffic through Charles.

## 4 Findings

### 4.1 API Testing

To map the API, we used a client device as described above and performed some typical user actions, such as logging in, viewing profiles, "liking" a user, and "passing" on a user. From these actions, we collated and tested a list of endpoints as described below.

**Authentication.** Authentication is the first step when connecting to the Tinder API. Other than the authentication (auth) endpoint, all requests to Tinder end-points require the X-Auth-Token ("Tinder Token") to be sent as a header in the request in the form of:

```
X-Auth-Token:   xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx*
```

* x is a hexadecimal digit, 4 is the literal number, and y is a character from the set {8, 9, a, b}.

If the Tinder token is not supplied, then Tinder returns a "401 Unauthorized" error message. Once the Tinder token has been set, requests can be made to all the other Tinder endpoints. Further, the Tinder token is the *one and only* object Tinder uses for verification, for all requests, following the initial authentication.

To receive a valid Tinder token, a user requires their Facebook ID as well as an appropriate Facebook token (as distinct from a Tinder token). The Facebook token is

generated by authenticating with Facebook through Tinder. The exact mechanics of Facebook authentication are out of the scope of this paper, and can be found on Facebook's developer site [17]. Additionally, Tinder requires other headers to be set which mimic a real phone's request. This includes any valid user agent, for example:

```
User-Agent: Tinder Android Version 4.5.2
```

In addition, the content type must also be set. This is in the form of:

```
Content-Type: application/json
```

We then used cURL[1] to create a valid Tinder authentication request:

```
curl -k -v -X POST -H "Content-Type: application/json"
-H "User-Agent: Tinder Android Version 4.5.2"
https://api.gotinder.com/auth --data
"{"facebook_token": "[Token]* ", "locale": "en",
"facebook_id": "[ID]* "}"
```

* Actual Facebook token and ID has been redacted for privacy reasons.

While the auth response also returned the authenticating user's complete profile information, our interest was the "token" field, which we set as our Tinder token. As the Tinder token is the only thing required to access Tinder's endpoints, we examined it more closely: if the token was predictable, then impersonating users would be possible by accessing the endpoints with other valid tokens. Having another user's token is extremely problematic because much of the authenticated user's information is available by accessing the profile end-point through:

```
curl -k -v -X GET -H "Content-Type: application/json" -
H "User-Agent: Tinder Android Version 4.5.2" -H ''X-
Auth-Token: xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx''
https://api.gotinder.com/profile
```

As you can see, this request only requires a valid Tinder token, and returns personal information including Facebook ID, date of birth, and exact GPS co-ordinates.

The tokens themselves are Version 4 UUID tokens and the document which defines them states:

> *"Do not assume that UUIDs are hard to guess; they should not be used as security capabilities (identifiers whose mere possession grants access), for example. A predictable random number source will exacerbate the situation* [18]*".*

While we personally found no problems with Tinder's usage of UUIDs, a theoretical attack, which would require intimate knowledge of a predictable random generator in use by Tinder, does exist. In essence, the attacker would first authenticate using their own account. Then, they would retrieve their randomly generated token, and

[1] cURL is a command-line utility to send and retrieve data over the Internet. Detailed documentation on history and usage is available at: http://linux.about.com/od/commands/l/blcmdl1_curl.htm (last accessed: 20/07/2016).

then use that token in the weak random number generator to generate the next valid token in the sequence. The attacker could then access the profile of, and see personal information of, the person who logged after them.

**Testing HTTPS.** One of the more basic things we wanted to check was that all sensitive traffic to Tinder was being sent over HTTPS. HTTP, which is not encrypted, is extremely vulnerable to man-in-the-middle attacks, as well as packet sniffing. Our examination revealed all traffic, except images, were sent over HTTPS (Fig. 7).

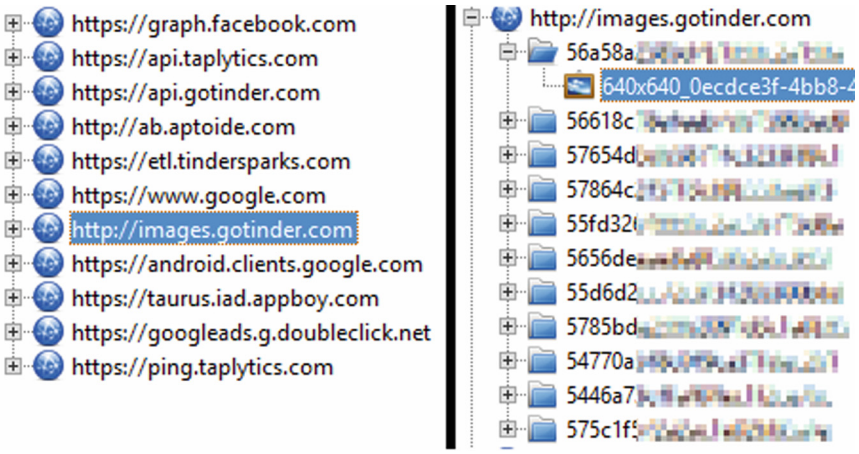


**Fig. 7.** Left: images.gotinder.com is not transmitted over HTTPS. Right: Tinder ID is visible in path of image URL (pixilated for privacy)

Pictures sent over plain HTTP can be intercepted through the methods mentioned above. We also found that a user's Tinder ID was transmitted in the path of the image, the relevance of which will become apparent when we discuss attacks further below. In addition, we found that the first image requested after authenticating was always the authenticated user. Being able to view images of people is not a security problem by itself, however users' belief that they are anonymous until they choose otherwise, may mean that they are prone to using more compromising images on Tinder. For example, there was an incident involving a married Canadian politician alleged to have made a Tinder account linked to his official campaign Facebook page [19]. Compounding this, users *could* be identified by using their image in a reverse image search. However, in our testing, this did not happen because most large web-crawlers respect Facebook's rules which disallow user profile crawling [20]. Should a user re-use their Tinder images elsewhere on the web, such as on other social media or dating websites, the probability for a successful reverse image search does increase. Further, if a user has an Instagram account connected, this again increases the chance that a user's image will be indexed on a search engine.

**Setting Location.** Setting the user's location was one of the potential privacy problems we wanted to investigate. While the previous trilateration attack involved creating

fake accounts [10]: physically moving a phone to all of these locations would have proven extremely time consuming. Indeed, most location-based attacks are only made feasible by being able to freely and rapidly adjust the attacker's location until ideal conditions are found. This is to say, being able to freely change your own location is a security concern. We found that adjusting your location through Tinder's API was trivial, as it can be accomplished through the following command:

```
curl -k -v -X POST -H "Content-Type: application/json"
-H "User-Agent: Tinder Android Version 4.5.2" -H "X-
Auth-Token:  xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx"
https://api.gotinder.com/user/ping --data "{'lat': 0.0*,
'lon': 0.0* }"
```

* A valid latitude and longitude are obviously required here.

As can been seen above, unlike some other geo-social apps, such as Mitalk or LOVOO [4], Tinder does not use checksums or signatures to validate the location it receives. Further, while Tinder does reject very small and very large changes to location, small changes can be accomplished by performing them in two opposed larger moves. Furthermore, very large changes can be performed by sending no location changes for a few minutes, and *then* sending the large change. We believe this behaviour is to accommodate users who turn off their phones and catch a plane, for example; otherwise these users may become permanently out of sync with Tinder.

### 4.2 Validating Previous Attacks

Now that all of the Tinder endpoints could be accessed, we wanted to verify that all of the old attacks that we examined earlier were actually patched, and no longer functional. The simplest method to test the first major vulnerability, which leaked exact user GPS data and Facebook ID, was to request a user's card and examine the results. We used a cURL command to accomplish this, in the form of:

```
curl -k -v -X GET -H "Content-Type: application/json" -
H "User-Agent: Tinder Android Version 4.5.2" -H "X-
Auth-Token:  xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx"
https://api.gotinder.com/user/000000000000000000000000*
```

* Substitution of dummy 0's with a valid Tinder ID is required.

Our response from Tinder is presented in Fig. 8:

As can been seen from Fig. 9, we confirmed that GPS co-ordinates and Facebook IDs are no longer being sent by Tinder. Furthermore, the distance returned is now a whole number, rounded to the nearest mile, rather than the exact distance we saw in the original trilateration attack [10]. However, we also wanted to test whether the distance returned was still accurate under specific circumstances, for example, if the distance was simply rounded to the nearest integer, gradually moving away from the target until the distance switched between two integers would give you exactly half the distance between those two integers, which you could then use as a point to conduct a

**Fig. 8.** Partial response from Tinder server to a user GET request. The Tinder ID field has been pixilated for privacy reasons

trilateration attack. We developed a program to try and accomplish this, which is detailed in the next section.

### 4.3 New Attacks

**Identifying Facebook users.** Our first attack came from noticing that an "ID" field is sent with some of Facebook "groups" data. As an example, examination of the information presented in Fig. 8 shows that ID "38078347904" was sent for our school details at University of South Australia. We then used this ID to visit the Facebook page for that group, and found everyone who liked the page. In this example, the URL would be:

`https://www.facebook.com/search/38078347904/likers`

Since we know that all Tinder users have a Facebook account and Tinder sources both name and photos from Facebook - we manually examined the list produced by the Facebook search until we found a user with a matching first name and similar image. We also had some success simply Googling the user's first name, along with their place of schooling, job, and any relevant bio data provided by Tinder. Although extremely basic, this attack did allow us to connect Tinder users and their Facebook accounts, which we consider a privacy problem. Also remember, that the only piece of information required for this attack to work is a Tinder ID, which, if you recall from our findings in Fig. 8, is sent as plain text and is not encrypted.

**Location attack.** We wanted to propose and test a new version of the previous trilateration attack. We developed a program that retrieved a Tinder ID from our matchmaking list and then moved our profile away slightly. We then requested the user's profile using their Tinder ID again and looked at whether Tinder reported their distance had changed. We picked 3 points, and did several refining passes on each to dial in the accuracy. The next figure demonstrates one of our results.

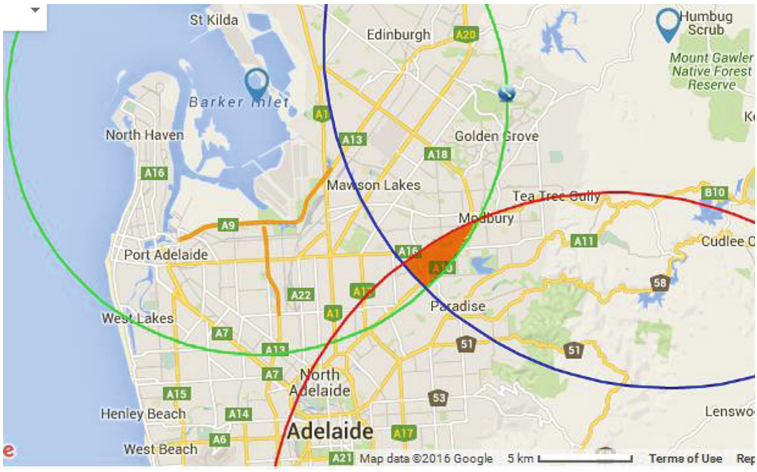


**Fig. 9.** Attempting our own trilateration when the reported distance changed from one whole number to the next produced a large 3–4 km$^2$ area of uncertainty (shaded orange). (Color figure online)

We tested several things, such as treating the integer returned as the floor, ceiling, as well as different rounding points – nothing seemed to produce anything more accurate than Fig. 9.

If the distance measured was not an accurate representation of the actual distance, then what was it? To test this we used two fake accounts, and had one "orbit" the other at a set distance as reported by Tinder. To do this, we used a reverse Haversine function that moved a set distance along a defined bearing. Again, we used several refining passes at our chosendistance (11 miles) so the point captured was exactly when Tinder changed from reporting 10 miles to 11 miles.

```
set bearingRads, set d //the bearing in radians to move
along, and the distance in metres to move

set φ1 //the latitude of the starting point in radians

set λ1 //the longitude of the starting point in radians

set R = 6371008 // average radius of the Earth

φ2 = Asin(Sin(φ1) * Cos(d / R) + Cos(φ1) * Sin(d / R) *
Cos(bearingRads))

λ2 = λ1 + Atan2(Sin(bearingRads) * Sin(d / R) *
Cos(φ1), Cos(d / R) - Sin(φ1) * Sin(φ2))

φ2 = toDegrees(φ2), λ2 = (toDegrees(λ2) + 540)%360-180
```

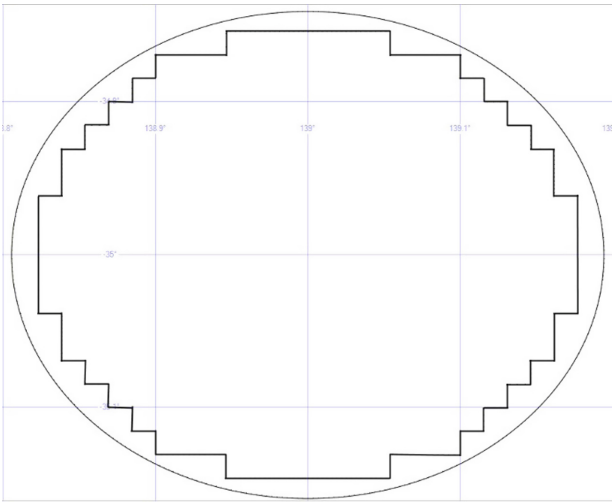

**Fig. 10.** Outer circle: an 11 mile (All of Tinder's internal representations of distance are in miles, which we have opted to keep for the sake of clarity) radius circle projected around a point. Inner shape: An 11 mile radius "circle" projected around the same point according to Tinder. Actual distance of inner shape to centre point ranged between 9.4883 and 10.6578 miles.

The function described above, looped over 360° produced the following:

From Fig. 10, it became obvious that Tinder has reduced accuracy in their distance calculation. We believe accurate GPS co-ordinates are being "snapped" to a less accurate grid which is then used for distance calculations performed by Tinder and displayed to the user.

If indeed the accuracy is being reduced at some stage of the process, it makes it extremely difficult, if not impossible to recover it.

While we were unable to perform an effective practical attack on the location of the user, we would like to propose the possibility of a further attack. Unfortunately, we were not able to determine if this attack would work, because pre-computing the Tinder grid over the internet was an extremely slow process, taking hours to get results within even one degree of accuracy. An attacker with considerable time, however, may gain success.

This attack would involve accurately precomputing the grid for Tinder (Fig. 11). The attacker would pick a GPS point, and move until they are snapped onto part of the pre- computed distance grid. Then, the attacker would manoeuvre to the very corner of the section they were in This would be Point 1. The attacker would then move to the left or right to find the reciprocal point on the other half of the grid. This would be Point 2. By measuring the distance from Point 1 to Point 2, and using a pre-computed table of distances, the attacker would now know which part of the grid they were on and be able to determine the bearing to the target centre point.

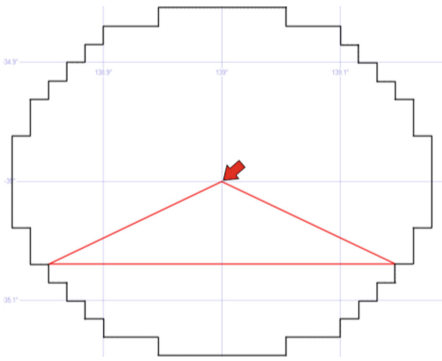

**Fig. 11.** An attack against the Tinder grid. This is an example 11 mile grid

Armed with two points, one distance, and two angles, the attacker could use trigonometry to find the final point, being the target (arrow on Fig. 11).

While this attack would only able to reveal which grid the target had snapped to, the grids appear to be $1/6^{th}$ of 0.1 degree of latitude or longitude at their most accurate. This equates to a square of roughly 2.8 $km^2$ (in South Australia), which is more accurate than the 3–4 $km^2$ identified in Fig. 9. Furthermore, it is unknown whether these grids are static, or whether their shape or accuracy can change.

## 5 Conclusion and Future Work

After significant probing we found that all known, significant, prior attacks on Tinder were no longer functional. We did find a number of security issues where Tinder has not followed best practice, including solely relying on UUID tokens for authentication, not properly validating location updates from users, sending user images over HTTPS, and including the Tinder ID in the user image path.

In terms of practical attacks, we found we were able to connect a Facebook profile to a Tinder account due to the shared information between them, the possibility of their images appearing on a search, as well as Tinder leaking the ID of certain Facebook groups the user was in. However, this attack was not consistent, required significant manual work, and depended on various luck factors such as Google indexing the right Facebook profile, or the user posting a Tinder image on the web.

For a location attack, we were able to get a rough location (3–4 $km^2$) using existing methods. We proposed a slightly more accurate attack ($\sim$2.8 $km^2$) using new methods, based on our belief that Tinder is now using grid snapping to remove exact user co-ordinates from calculations. However, owing to the apparent extremely inefficient pre- calculations required, we were unable to complete the attack ourselves.

Future work includes extending the research to grid-based attacks to speed up the attacks, and to attempt a practical application.

## References


1. Tinder. https://www.gotinder.com/press. Last accessed: 20 July 2016
2. Russel, S., Kissick, D.: Is Tinder really creating a 'dating apocalypse'? https://www.theguardian.com/technology/2015/aug/16/tinder-app-creating-dating-apocalypse-twitter-storm. Last accessed: 20 July 2016
3. Farnden, J., Martini, B., Choo, K-K.R.: Privacy risks in mobile dating apps. In: 21st AMCIS 2015 (2015). http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1427&context=amcis2015. Last accessed: 21 July 2016
4. Zhao, S., Luo, X., Bai, B., Ma, X., Zou, W., Qiu, X., Au, M.H.: I know where you all are! exploiting mobile social apps for large-scale location privacy probing. In: Liu, J.K.K., Steinfeld, R. (eds.) ACISP 2016. LNCS, vol. 9722, pp. 3–19. Springer, Cham (2016). doi:10.1007/978-3-319-40253-6_1
5. Tinder on Google Play Store. https://play.google.com/store/apps/details?id=com.tinder. Last accessed: 17 July 2016
6. I found out my internal Tinder rating and now I wish I hadn't. https://www.fastcompany.com/3054871/whats-your-tinder-score-inside-the-apps-internal-ranking-system. Last accessed: 20 July 2016
7. Official Tinder Blog – Giphy Announcement. http://blog.gotinder.com/say-more-with-tinders-new-messaging-features/. Last accessed: 20 July 2016
8. Official Tinder Blog – Super Like Announcement. http://blog.gotinder.com/updated-introducing-super-like-a-new-type-of-swipe/. Last accessed: 21 July 2016
9. Tinder Privacy Breach. http://qz.com/107739/tinders-privacy-breach-lasted-much-longer-than-the-company-claimed/. Last accessed: 20 July 2016
10. Trilateration attack on any Tinder user. http://blog.includesecurity.com/2014/02/how-i-was-able-to-track-location-of-any.html. Last accessed: 20 July 2016
11. Tinder Bot "Bonfire". https://www.tinderliker.com/. Last accessed: 20 July 2016
12. Tinder Bot "Auto Liker". http://tinderautoliker.com/. Last accessed: 20 July 2016
13. Tinder Bot with EigenFaces. http://crockpotveggies.com/2015/02/09/automating-tinder-with-eigenfaces.html. Last accessed: 20 July 2016
14. Official Symantec Blog – Tinder Spam. http://www.symantec.com/connect/blogs/tinder-spam-year-later-spammers-still-flirting-mobile-dating-app. Last accessed: 20 July 2016
15. Official Tinder Blog – Limited Swipe Announcement. http://blog.gotinder.com/keeping-tinder-real/. Last accessed: 20 July 2016
16. D'Orazio, C., Choo, K.-K.R.: An adversary model to evaluate DRM protection of video contents on iOS devices. Comput. Secur. **56**, 94–110 (2016)
17. Facebook Developer Site – Access Tokens. https://developers.facebook.com/docs/facebook-login/access-tokens/. Last accessed: 20 July 2016
18. A Universally Unique IDentifier (UUID) URN Namespace. https://tools.ietf.org/html/rfc4122. Last accessed: 20 July 2016
19. Married Politician on Tinder. http://www.news1130.com/2015/01/12/married-lower-mainland-federal-politician-appears-on-dating-app-tinder/. Last accessed: 20 July 2016
20. Facebook Robots File. https://www.facebook.com/robots.txt. Last accessed: 20 July 2016