Can we safely learn about users' passwords

What we should never know

Jeffrey Goldberg
jeff@1Password.com

August 9, 2020
(Last revised: August 11, 2022)

1Password

Setting the scene

- Jeffrey Goldberg
- Working at 1Password since 2010
- Wants to know everything
- So committed to Zero-Knowledge that I crave ignorance

- A password manager. Client software and a service.
- The client software is where most of the action is.
  - Unlocked client knows names of vaults and names of items. Server doesn't.
  - Unlocked client knows password strengths. Server doesn't.
  - Unlocked client has decrypted encryption keys. Server doesn't.
- Designed so that we learn as little about users secrets as possible
- We like to think we know a thing or two about password behavior

1. What can we learn about 1Password user's behavior without putting them at risk?
2. What technologies for doing so are within our reach?

Setting the scene

The example behavior question

- If we can figure this out for extremely sensitive information then we can do it for anything.
- I pick a real example question about user behavior that involves understanding extremely sensitive data.

The use of MFA [for 1Password itself] may lead people to use weaker [account] passwords, thereby strengthening a less crucial part of their security (authentication), while weakening a far more important component. [Gol18]

Data question

Is there a negative correlation between use of 2FA for 1Password itself and the strength of a the account password?

1. We know who has 2FA switched on. (This is necessary to provide the service.)
2. We don't know the password strength of anybody's account password.

For user eyes only

We don't ever want to know the strength of anyone's account password.

(I don't think I really need to list the reasons.)

Not knowing is not enough. The world needs to know that we don't know.

- Each participating user should be able to determine that their privacy is being protected.
- The Security and Privacy communities must be able to confirm that our system behaves as we say it does.

Today's talk is an over view of what I have learned so far about how we might do this.

Much of what I learned is new or new-ish to me. But there is nothing fundamentally new here. So you can spend the remainder of this talk napping.

Overview of potential techniques

- Clients have the sensitive data.
- Clients do something to that data and send it to a system that is not under the user's control.

Overview of potential techniques

Anonymization

- Clients will not send any identifying data.
- Clients can't avoid "sending" IP addresses. (Tor might help with that)

- Anonymization is hard, but even when done right it isn't enough.
- Information wants to be free! (Assume your data will leak)
- Anonymized data combined with other (public) data can be de-anonymized.
- That other public data can come from sources you have nothing to do with.
- That other data might not yet exist at the time you create your scheme.

(Talk about Netflix de-anonymization if time)

Good anonymization and data protection is necessary. But they are far from sufficient.

Overview of potential techniques

Vague response

The client shouldn't report password strength with the full precision that it knows, but could, say, use three bins: low, medium, high.

Overview of potential techniques

Random response

In one setup [BBB79] Subjects given a questionnaire but told to roll a die before answering each question. The instructions told them to answer differently depending on their roll of the die

1–4 Answer honestly

5 Answer "yes"

6 Answer "no"

More recently subjects drew a red or green ball out of a sack and answered the red or green question. [Lar+06]

Red "Did you ever interrupt a pregnancy?"

Green "Were you born in April?"

This was conducted in Mexico in 2001. Abortion was highly stigmatized and illegal.

Researchers have been able to demonstrate that the increased rate of honest responses outweighs the statistical noise.

Suppose we set up a system to answer honestly about password strength 50% of the time. If our data leaks and is de-anonymized is the user sufficiently protected.

- Vague response reduces the power of statistical tests.
- Random response reduces the power of statistical tests.
- Opt-in reduces the sample size and introduces a selection bias.

(I have started to play with simulations to see what kind of samples and parameters are likely to still produce usable results)

Differential privacy

- DP is about limiting the possibility of de-anonymization
- DP provide a common mathematical notion of dataset privacy protections across a wide variety of techniques
- It adds statistical noise in ways similar to vague and random response
- The noise it adds is designed to make it possible to see its effect on privacy

- …a single technique
- …applied at a common point of data processing
- …particularly easy.

Different DP techniques can be done at

- at data collection time
- to transform a dataset
- at data analysis time

In our example, we never want to have password strengths so it would have to be done client side at data collection time.

Differential privacy

Homomorphic encryption

Two millionaires wish to know who is richer; however, they do not want to find out inadvertently any additional information about each other's wealth. How can they carry out such a conversation? [Yao82]

- For any protocol that can run in polynomial time using a trusted third party, there is a protocol that can produce the same results without a TTP.
- Allows multiple parties to compute things over their individual secrets without revealing secrets to each other or a third party.

- Still need a PhD to make real use of it?
- Polynomial time and space doesn't actually mean fast and small.
- Protocols have a lot of back and forth.

A rant on noise

- These techniques add statistical error
- They add known, quantifiable amounts of statistical error
- Thus you either need a larger sample, or you have wider confidence intervals in your results
- Some objections have been "we can't accept anything with statistical error or confidence intervals."

These techniques add known amounts of error on top of the error that one would *already have* without these techniques.

If you can't handle statistical error or you don't know that you always have some you should not call yourself a data analyst.

Conclusions?

Is there a combination vagueness, random responses, and data protections on the acquired data that would offer sufficient guarantees for our users and allow us to answer the 2FA/strength question with sufficient confidence?

When I started, I hoped my math was good enough to figure this out analytically. Now I must resort to simulations.

NETFLIX CASE

- In 2006 Netflix publicly released an anonymized dataset of about 100 million movie ratings from about 480,000 subscribers.
- From Netflix FAQ "all customer identifying information has been removed; all that remains are ratings and dates."

- Some Netflix subscribers publicly share some movie ratings in other places.
- Some of those people may not want to world do know what about some of the other movies they watched.

Narayanan and Shmatikov [NS08] use Netflix and IMDB data to illustrate a general algorithm for de-anonymization:

> *With 8 movie ratings [from IMDB] (of which 2 may be completely wrong) and dates that may have a 14-day error, 99% of records [can] be uniquely identified in the dataset. For 68%, two ratings and dates (with a 3-day error) are sufficient.*

> *Attacks only get better with time. The flood of de-anonymization demonstrations in the last decade makes for a strong argument that database privacy should rest on provable guarantees rather than the absence of known attacks. The flourishing research on differential privacy is thus a welcome development.* [NS19, p. 1]

And there have been a flood of de-anonymization demonstrations.

Resources

- Repository containing these slides (and a few other things): `https://gitlab.com/1Password/ppa`
- OpenDP: Really nice tools and community for the DP at the data analysis stage. `https://opendp.org`
- Usenix PEPR: Privacy Engineering, Practice and Respect.

[BBB79]   Guy Bégin, Michel Boivin, and Jeannette Bellerose. "Sensitive Data Collection Through the Random Response Technique: Some Improvements." In: *The Journal of Psychology* 101.1 (1979), pp. 53–65. DOI: `10.1080/00223980.1979.9915052`. eprint: `https://doi.org/10.1080/00223980.1979.9915052`. URL: `https://doi.org/10.1080/00223980.1979.9915052`.

[Gol18]   Jeffrey Goldberg. "What does "MFA" mean?" In: *PasswordsCon SE*. PasswordsCon. Nov. 2018. URL: `https://blog.1password.com/files/what-the-secret-key-does/goldberg-mfa.pdf`.

[Lar+06]   Diana Lara et al. "The Measure of Induced Abortion Levels in Mexico Using Random Response Technique." In: *Sociological Methods & Research* 35.2 (2006), pp. 279–301. DOI: `10.1177/0049124106290442`. eprint: `https://doi.org/10.1177/0049124106290442`. URL: `https://doi.org/10.1177/0049124106290442`.

[NS08]   Arvind Narayanan and Vitaly Shmatikov. "Robust de-anonymization of large sparse datasets." In: *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE. 2008, pp. 111–125. DOI: `10.1109/SP.2008.33`.

[NS19]   Arvind Narayanan and Vitaly Shmatikov. *Robust de-anonymization of large sparse datasets: a decade later*. May 12, 2019. URL: `https://www.cs.princeton.edu/~arvindn/publications/de-anonymization-retrospective.pdf`.

[Yao82]   Andrew C Yao. "Protocols for secure computations." In: *23rd annual symposium on foundations of computer science (sfcs 1982)*. IEEE. 1982, pp. 160–164.