This is going to be a long one – sorry! I find cryptology fascinating. On the second article regarding hashing for anonymization. The author makes the point that a semi-capable technical person could reverse engineer the hash function to find the input value. Not by finding the mathematic inverse but simply running the hash function to determine the right key.

According to this article, <https://www.synopsys.com/blogs/software-security/cryptographic-hash-functions/>, Ed Felten’s comments are not only accurate, but paint a rosy picture. Not only can hashed information be broken into via “brute force” but several hashing algorithms can output the same hash value from different input values. And to make things even worse, there are optimized lookup tables called “rainbow tables” which are precomputed ctables for caching the output of cryptographic hash functions (<https://en.wikipedia.org/wiki/Rainbow_table>).

The first comments on the Ed Felten article talked about using salt values as a further security setting. A salt value basically adds a random string of characters to the end of the input value, before putting them into a hash function. So *password* becomes *password%KJNFO6e98r55\*\**  before being thrown into the hash function. I have to agree with a later commenter, Dan Kaminsky, who said that someone who could attack against hashes is also able to do so against salted hashes as well.

It turns out Dan Kaminsky was a computer prodigy (taught himself to code at the age of 5 in 1984) and broke into a military computer system at the age of 11. In 2008 he discovered the DNS Spoofing hack (<https://en.wikipedia.org/wiki/DNS_spoofing>) which is still ongoing today despite governments and corporations working on solving the problem.

Now, the question for this discussion is how to balance privacy along with the desire to be transparent? I don’t really see a desire for transparency. In most instances, the consequences of anonymizing the data will not impact the consequences of the results. To use the assignment for a basic example, I anonymized the information on the graph and changed all of the words to animals. The graph didn’t necessarily make sense - the correlation between Tarsiers on the x-axis and Elephants on the y-axis with color groups of 5 other animals – but it doesn’t necessarily have to. Even with the strange legend, the impact of the graph remained the same; and in the real world, we could use better dummy variables.

Even when there seems to be a need for actual transparency, pharmaceutical research comes to mind, anonymization still takes precedence with double blind studies, or dynamic data masking to use a term in the lecture. Now the method of anonymization becomes the key (cryptographic pun intended). Because doing a random data shift, such as the epsilon privacy, could lead to drastic issues in analysis when the drug is ready to go before the FDA. I would think the best way to retain anonymity is a multi-step, multi-function privacy protocol such as:

1. Someone assigns a randomized salt value that differs dynamically (16 characters on one, 8 on the other, 12 on the third, 12 on the 14th, etc.)
2. Someone performs k-anonymity as appropriate for the situation.
3. Someone furthers k-anon by using L-Diversity.
4. Someone analyzes the data.

Something I do wonder about is the possible loss of accuracy as privacy security measures are implemented.