Differential Privacy

There are situations where Apple can improve the user experience by getting insight from what many of our users are doing, for example: What new words are trending and might make the most relevant suggestions? What websites have problems that could affect battery life? Which emoji are chosen most often? The challenge is that the data which could drive the answers to those questions—such as what the users type on their keyboards—is personal.

Apple has adopted and further developed a technique known in the academic world as *local differential privacy* to do something really exciting: gain insight into what many Apple users are doing, while helping to preserve the privacy of individual users. It is a technique that enables Apple to learn about the user community without learning about individuals in the community. Differential privacy transforms the information shared with Apple before it ever leaves the user's device such that Apple can never associate the data it receives to any individual user

The differential privacy technology developed by Apple is rooted in the idea that statistical noise that is slightly biased can mask a user's individual data before it is shared with Apple. If many people are submitting the same data, the noise that has been added can average out over large numbers of data points, and Apple can see meaningful information emerge.

Privacy budget

Because there is a slight bias to the noise that is added to contributions made using these local differential privacy techniques it is theoretically possible to determine information about a user's activity over a large number of observations from a single user (though it's important to note that Apple doesn't associate any identifiers with information collected using differential privacy). For that reason, the Apple differential privacy implementation incorporates the concept of a per-donation *privacy budget* (quantified by the parameter epsilon), and sets a strict limit on the number of contributions from a user in order to preserve their privacy.

Apple uses local differential privacy to help protect the privacy of user activity in a given time period, while still gaining insight that improves the intelligence and usability of such features as:

- QuickType suggestions
- Emoji suggestions
- Lookup Hints
- Safari Energy Draining Websites (iOS 11)
- Safari Autoplay Intent Detection (iOS 11)
- Safari Crashing Websites (iOS 11)
- Health Type Usage (iOS 11)

For each feature, Apple seeks to make the privacy budget small while still collecting enough data to to enable Apple to improve features. Apple retains the collected data for a maximum of 18 months. The donations do not include any identifier, and IP addresses are not stored.

For Lookup Hints, and QuickType, Apple uses a privacy budget with epsilon of 4, and limits user contributions to two donations per day. For emoji, Apple uses a privacy budget with epsilon of 4, and submits one donation per day.

For Health types, Apple uses a privacy budget with epsilon of 2 and limits user contributions to one donation per day. The donations do not include health information itself, but rather which health data types are being edited by users and with what frequency.

For Safari domains identified as causing high energy drain, Apple uses a single privacy budget with epsilon of 4 and limits user contributions to 2 donations per day. For Safari Autoplay intent detection, Apple uses a privacy budget with epsilon of 4 and limits user contributions to 2 donations per day.

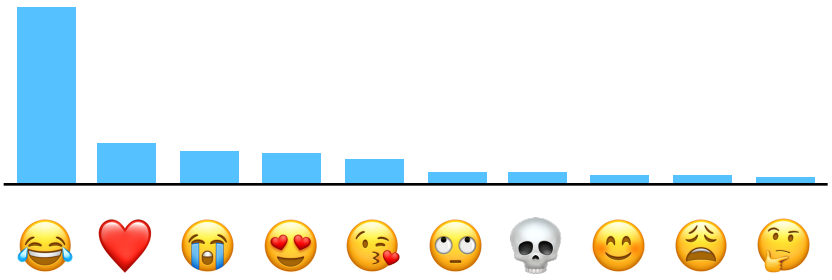
Techniques

Local differential privacy guarantees that it is difficult to determine whether a certain user contributed to the computation of an aggregate by adding slightly biased noise to the data that is shared with Apple. But before adding this noise, it's necessary to define a data structure that captures a sketch of user input with a small number of bits. Apple currently makes use of two specific techniques:

Count Mean Sketch

In our use of the Count MeanSketch technique for differential privacy, the original information being processed for sharing with Apple is encoded using a series of mathematical functions known as *hash functions*, making it easy to represent data of varying sizes in a matrix of fixed size.

The data is encoded using variations of a SHA-256 hash followed by a privatization step and then written into the sketch matrix with its values initialized to zero.



The Count Mean Sketch technique allows Apple to determine the most popular emoji to help design better ways to find and use our favorite emoji. The top emoji for US English speakers contained some surprising favorites.

The noise injection step works as follows: After encoding the input as a vector using a hash function, each coordinate of the vector is then flipped (written as an incorrect value) with a probability of $1/(1 + e^{\varepsilon})$, where ε is the privacy parameter. This assures that analysis of the collected data cannot distinguish actual values from flipped values, helping to assure the privacy of the shared information.

In order to stay within the privacy budget we do not send the entire sketch matrix to the server but only a random row of the matrix. When the information encoded in the sketch matrix is sent to Apple, the Apple server tallies the responses from all devices sharing information and outputs the mean value for each element of the array. Although each submission contains many randomized elements, the average value across large numbers of submissions gives Apple meaningful aggregate data.

Hadamard Count Mean Sketch

The Hadamard Count Mean–based Sketch technique uses a noise injection method similar to the one used in the Count Mean Sketch technique, but with an important difference: It applies a type of mathematical operation called a Hadamard basis transformation to the hashed encoding before performing the privatization step. Additionally, it samples only 1 bit at random to send instead of the entire row as in the Count Mean Sketch technique. This reduces communication cost to 1 bit at the expense of some accuracy.

Seeing user data

Users can examine the information being shared with Apple for the categories of data that are protected using Differential Privacy. In iOS, the information is visible under Settings > Privacy > Analytics > Analytics Data, under the Differential Privacy category. In macOS, users can launch the Console app and view the information under the Differential Privacy category of System Reports.

Controlling participation

The data-gathering features that use differential privacy in macOS High Sierra and iOS 11 are linked to the user setting for Device Analytics. Users are presented with the option of sending diagnostic information when they set up a device running macOS or iOS, and they can always change their choice later in System Preferences on macOS or the Settings app on iOS.

The beginning

Apple launched differential privacy for the first time in macOS Sierra and iOS 10. Since then, we have expanded to other use cases such as Safari and Health types. As Apple continues to refine differential privacy algorithms, we look forward to using them to improve user experience in other areas of our products, while continuing to work to protect our users' private information.