1 System component

Differential privacy framework located at **/System/Library/Private Frameworks/DifferentialPrivacy.framework**, contains code to implement differential privacy framework.

Daemon **com.apple.dprivacyd**, located at **/usr/libexec/dprivacyd**, is a daemon handling differential privacy.

Database located at /private/var/db/DifferentialPrivacy, contains several tables with privatized data and a table related to available budget per record type.

Configuration files located at **/System/Library/DifferentialPrivacy/C onfiguration/**, four files, contain numerical parameter to configure the action of differential privacy framework

Report file located at **/Library/Logs/DiagnosticReports/** and **/private /var/db/DifferentialPrivacy/Reports/**, contain perturbed data and ones transmitted to Apple's servers

2 System organization

The use of DP focuses on four applications: emoji, new words, deep link, loop up hints in Note.

Whenever user enter a emoji or a new word in Note, the datum is perturbed by differential privacy algorithm and the privatized version is stored in corresponding database table.

ReportGenerator task runs periodically to select record from tables and write them into report files, it also marks selected reords as ”submitted”, and finally delete them by other periodically tasks.

3 System's Details

3.1 Database

There are two tables to store perturbed data, ZOBHRECORD and ZCMSRECORD, the former dedicated to the emoji datum, and the latter dedicated to only words which is not previously typed.

Another table is ZPRIVACYBUDGETRECORD, which contains 7 entries, one for each of applications(NewWords, Emoji, AppDeepLink, Search, and health) and the other two for helper functions(default and testBudget)

3.2 Configuration files

There are 4 configuration files for the daemon "com.apple.dprivacyd". They are

1. $keynames.plist$
2. $keyproperties.plist$
3. $algorithmparameters.plist$
4. $budgetproperties.plist$

keynames.plist contains a mapping of keyname to keyproperties. And each keyname is assigned 13 possible keyproperties.

For each of the keyproperty, keyproperties.plist specifies a privatization algorithm and privatization parameter. Then, for each propertyname, keyproperty.plist specifies a budgerkeyname. And in particular, for all newwords, regardless of language, have the same budgetkeyname "com.apple.keyboard.NewWords", and for all emoji, have the same budgetkeyname "com.apple.keyboard.Emoji" and such. There are total 7 distinct budgetkeyname which is consist with ZPRIVACYBUDGETRECORD table.

Algorithmparameter.plist specifies additional parameter of privatization parameter.

budgetproperties.plist specifies two quantifies for each of budgetkeyname: sessionseconds and sessionamount.

ReportGenerator task select record from database tables to be included in the record file according to two rules:

1. Select most min records per Keyname.

2. The number of records in the same BudgetKeyName may not exceeds available privacy balance, if the total number of record exceeds available privacy balance, ReportGenerator task choose subset of records whose number doesn't exceed available privacy balance randomly.

PrivacyBudgetMaintenance task increase privacy budget for each BudgetKeyName according to corresponding SessionAmount and SessionSecond. That is, the task increase privacy budget by SessionAmount every SessionSecond. When user opts-in to differential privacy data collection, privacy budgets for each BudgetKeyName are initialized by SessionAmount.

4 protection

Without root permission:

1. The configuration files are difficult to change because of Apple's System Integrity Protection in Mac OS.

2. Even if one success in changing configuration files, the privacy parameter is hard-coded in the code of framework

3. Time measures are hard-coded in the code, so it's useless to change SessionSecond or system clock to accelerate privacy budget increase or report file generation.

5 Conclusion

Some shortcomings: 1. Apple doesn't explain privacy loss, which violates concept of differential privacy that use can know their privacy loss by data collection.

2. Apple may abuse user's privacy data intentional or unintentional. It can be realized by introducing more BudgetKeyName and changing other parameters.

3. The privacy loss of 16 is higher than what is considered in literature, and the privacy parameters update periodically can increase privacy loss if you don't use some application for some time.

4. The implementation leaks features of MACs a user is using and in what language and with what keyboard preference and such. And because only new words can be pri-

vatized and added to relative database table, one can test whether a word has ever been used by observing whether typing it triggers changes to differential privacy database.