Extending the idea of access and privilege found in typical file systems might yield surprising benefits in solving the thorny problem of determining on how data found in public records should be treated on the spectrum from open/public to closed/private. Just as privileges are assigned based on the identity of the user in a file system, perhaps a similar approach can be adopted for specific data within a dataset.

This could be accomplished with two different techniques: 1) complex data types for metadata could be defined that specify, on a 0-1 scale, a privacy utility for the data, data defined as 'private' would have a utility as an attribute and all types of private data would specialize the value of utility, and 2) these types could be mapped to access control that allows various levels of access to the data. To accomplish #1, an automated system could be constructed to process your existing data and compute a privacy utility for individual elements of blocks of data. For instance, let's assume SSN's or Bank Account data represent a privacy utility of 1 in a highly risk averse utility function for privacy. For blocks of text, write a Python program with regular expressions that match SSN's and Bank Account routing and account data. For each instance that matches the expression, annotate the data with a metadata tag for the class of private.SSN or private.bankAccount and assign it a utility based on a metadata class utility assignment, e.g. private.SSN.utility ==1.

As a user navigates the database, these privacy utilities are matched to the access privileges of the user. User.public would only have a privacy access privilege of 0 utility. A simple approach would be to automatically redact that information as the user navigates the data over the privacy 'landscape'. Unless the user has superuser privileges, like a data administrator, the information simply disappears from reading privilege. This approach would allow relevant but not private information to be displayed to the public and would satisfy the objective of informing the public without compromising privacy.

For example, imagine a group of students has been awarded grants from the City of Seattle and their name, address, and SSN information is in the database. When this data is accessed, the redactor generates a readable copy of the data modifying the data based on the aforementioned privacy utilities and access control so that:

user.public: no view, redacted
user.public.cityemployee: names (read), address and SSN (redacted)
user.public.cityemployee.programadministrator: name, addresses (read), SSN (redacted)
user.public.cityemployee.administrator.IT.dbadministrator: name, addresses, SSN(read)

User.public would read a record that omits the private data of the students while the User..ITdbadminstrator would have full access to all the data in order to properly administer it, e.g. change an address.

Since the concept of risk, utility, and expected value have already been introduced in the Draft Report, none of these ideas will be exceptionally novel and the idea of metadata tags and access control are well understood.

I hope the basic idea of combining some of these well known approaches in the context of privacy can be of some value to this difficult problem. A side benefit might be to cache the behavior of users accessing the data - for instance users who consistently browse data outside their privilege, might have their data surfing habits scrutinized more closely than the average user who might stumble across data on a relatively infrequent basis. Capturing the transactional stream might also have the positive benefit of re-evaluating the privacy settings on data to provide public access in areas that are learned to be of little privacy exploitation value.

Best Regards,

- Karl Keller