



HHS WORKSHOP ON THE HIPAA PRIVACY RULE'S DE-IDENTIFICATION STANDARD

MARCH 8–9, 2010
MARRIOTT AT METRO CENTER, WASHINGTON, DC

De-Identification

Jules J. Berman, Ph.D., M.D.

Panel #: 1, March 8





- Reasons for De-Identification
 - Exchange and combine large, complex data sets containing human subject info, from multiple sources.
 - Conduct human subject research without harming patients
 - Avoid impossible task of getting individual informed consent on thousands/millions of records
 - Exempt from HIPAA regulations
 - Potentially exempt from Common Rule regulations for human subject research
 - Make vital contributions to medical science, provide better health care, protect nation's health (as per HITECH)





- Historical development of de-Identification under Common Rule and HIPAA
 - In the Common Rule (45 CFR 46, Protection of Human Subjects, 1991) there was the general concept if you remove all links to the person, the data becomes disembodied, and thus safe
 - Not focused on combining data from multiple sources, accruing data over time, or revisiting source data.
 - Not always safe
 - De-identification (HIPAA Privacy Rule, 2003) permits re-identification and provides a narrow opportunity to bind a record to a unique object without harming patients





- Processes in dataset De-Identification
 - Removing identifiers from individual records (**most attention but least difficult part of job**)
 - Making sure there are no records with unique sets of data that can identify an individual (ambiguation)
 - *Providing a unique code to each data record that will be the same for every record belonging to an individual (**most difficult part of job**)
 - *Data scrubbing (removing private information from text)





- Providing a unique code to each data record that will be the same for every record belonging to an individual and will not be used for other individuals
 - Sometimes confusingly referred to as providing a unique identifier for the record
 - **John Q. Public “Glucose 85” “9/20/03”**
 - Replace with one-way hash performed on John Q. Public (name cannot be computed from hash value)
 - **93828506069828474 “Glucose 85” “9/20/03”**





- **93828506069828474 “Glucose 85” “9/20/03”**
- John Q. Public comes back about a year later and has another glucose test.
- **John Q. Public “Glucose 93” “10/15/04”**
- Perform one-way hash on John Q. Public
- **93828506069828474 “Glucose 93” “10/15/04”**
- Combine your data
- **93828506069828474 “Glucose 85” “9/20/03”**
- **93828506069828474 “Glucose 93” “10/15/04”**





- Problem: **Vulnerable to dictionary attack**
- Explicitly forbidden under HIPAA to achieve de-identification of a dataset by using a one-way hash on an identifier.





- **Solution: Use a **zero-knowledge protocol** to determine if two records belong to the same person**
 - A zero-knowledge protocol is a way of resolving a question without learning anything about the subjects in question, other than the answer to your question.
 - A patient's identifier is added to a random number, producing a new random number, for the two records being compared; if the new random number is the same for both, the patient in both records is the same
 - If so, assign both records the same unique random code (e.g., uuid). HIPAA and Common Rule exempt.





- A big problem assigning a unique code to each record is the absence of a **national patient identifier system** in the U.S.
 - Without national patient identifier, you've got to settle for flawed alternate methods (name, social security number, social security number plus birthdate).
 - The weakness in many EHR systems is poor patient identification (one patient with multiple identifiers, one identifier with multiple patients)





- Data scrubbing: removing private information, including identifiers, incriminating and embarrassing remarks, information unrelated to the necessary use of the data
 - Applies to free-text data





- Data scrubbing: Two methods
 - **One way:** Remove everything from the data that is found on a list of forbidden words and phrases.
 - **Another way:** Remove everything from the data that is absent from a list of acceptable phrases.





- Remove everything from the data that is found on a list of forbidden words and phrases
 - Produces a readable output, but slowly.
 - Requires an up-to-date list of bad words and phrases (patient names, staff names, etc.)
 - **Reduces identifiers, does not eliminate all identifiers**
 - **To the best of my knowledge, has never been used to prepare de-identified data to the public.** Used in “Data Use Agreements” - no scientific value because not publicly reviewed or shared





- Remove everything from the data that is absent from a list of acceptable phrases
 - Removes all identifiers if the list is clean.
 - Can be used to make public release data.
 - Very fast (thousands of times faster than alternate method).
 - Simple code, in public domain. (Can be implemented in 16 lines).
 - **Chief drawback: Provides an inferior output with respect to readability**





- **Advice regarding de-identification under HITECH**
 - In absence of national patient identifier system, the most important task of Information systems is to provide a unique identifier to each patient. There should be certified public protocols to accomplish this and EHR certification should focus on this task.
 - Certify public methods for comparing patient records across institutions to determine when two records belong to the same patient (as per zero knowledge protocol).





- **Advice regarding de-identification under HITECH**
 - Certify public methods that bind a unique number to a patient record (to aggregate records across institutions and across time).
 - Certify public protocols, algorithms, and software routines that scrub free-text data.





- **Advice regarding de-identification under HITECH**
 - **Medical informatics is a serious field of study, and can't be mastered by attending a few meetings. Guidelines for curricula that include in-depth discussions of the issues covered in these panels should be written and distributed to educational facilities.**





• References to some of my works

- Berman JJ. Confidentiality for Medical Data Miners. *Artificial Intelligence in Medicine*. 26(1-2):25-36, 2002.
- Berman JJ. Threshold protocol for the exchange of confidential medical data. *BMC Medical Research Methodology*. 2:12, 2002.
- Berman JJ. Concept-Match Medical Data Scrubbing: How pathology datasets can be used in research. *Arch Pathol Lab Med*. 127:680-686, 2003. (Concept-Match has been replaced by doublet method, see Ruby Programming book)
- Berman JJ. Zero-Check: A Zero-Knowledge Protocol for Reconciling Patient Identities Across Institutions. *Archives of Pathology and Laboratory Medicine* 128:344-346, 2004.
- Berman JJ. Nomenclature-based data retrieval without prior annotation: facilitating biomedical data integration with fast doublet matching. *In Silico Biology* 5, 0029, 2005.
- Berman JJ. *Biomedical Informatics*. Jones and Bartlett, Sudbury, MA, 2007.
- Berman JJ. *Perl Programming for Medicine and Biology*. Jones and Bartlett, Sudbury, MA, 2007.
- Berman JJ. *Ruby Programming for Medicine and Biology*. Jones and Bartlett, Sudbury, MA, 2008.
- Berman JJ. *Methods in Medical Informatics: Fundamentals of Healthcare Programming in Perl, Python, and Ruby*. CRC Press, Chapman & Hall/CRC Mathematical & Computational Biology, 2010
- Web site with links to programs and text of papers: <http://www.julesberman.info/>
- blog site: <http://julesberman.blogspot.com/>

