HIDE: Short IDs for Robust and Anonymous Linking of Users Across Multiple Sessions in Small HCI Experiments

Frode Eika Sandnes*

Oslo Metropolitan University, Oslo, Norway

Record linking is needed to analyze observations across multiple sessions. However, recent privacy legislature such as the General Data Protection Regulations (GDPR) restricts the storage of information that identify individuals. Obtaining permissions to store information about individuals can be bureaucratic and time-consuming. Anonymous schemes such as self-generated ids and machine generated ids have been proposed. However, self-generated linking ids demand effort from the participants, while machine assisted schemes typically generate long and incomprehensible ids. Consequently, there is a risk that students and researchers will limit their research to single session experiments to avoid privacy issues. To simply the administration of small multi-session experiments, the HIDE procedure is proposed for generating short human readable ids for linking participants across multiple sessions while maintaining anonymity and being robust to input errors. The approach is different from previous approaches in that the goal is to minimize the length of the linking ids. First, the procedure converts the participant's name into a phonetic representation. Next, this phonetic representation is hashed, and a truncated snippet of the hash is used as the linking id. HIDE is initialized by searching for a salt that minimizes the id lengths. Experiments show that the procedure is capable of coding small experiments with 20 participants using two digits, and experiments of around 200 participants with four digits. An implementation of the procedure has been made available through a simple web interface. It is hoped that the procedure can help students and HCI researchers collect more comprehensive data by following participants over time, while protecting their privacy.

CCS CONCEPTS • CCS Human-centered computing → Human computer interaction (HCI) → Empirical studies in HCI

Additional Keywords and Phrases: Longitudinal studies, Multi session experiments, Record linking, Soundex, Hashing, Privacy, GDPR

ACM Reference Format:
First Author's Name, Initials, and Last Name, Second Author's Name, Initials, and Last Name, and Third Author's Name, Initials, and Last Name. 2018. The Title of the Paper: ACM Conference Proceedings Manuscript Submission Template: This is the subtitle of the paper, this document both explains and embodies the submission format for authors using Word. In Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY. ACM, New York, NY, USA, 10 pages. NOTE: This block will be automatically generated when manuscripts are processed after acceptance.

1 INTRODUCTION

To get a deeper understanding of new interactive technologies, it is necessary to follow participants over time to compare first time use with use after practice. Longitudinal experiments, pre-test/post-test experiments, and other experimental designs involving multiple sessions are common in HCI [19, 31, 46], for example, for evaluation of

* Place the footnote text for the author (if applicable) here.

learning over time with new text entry methods [35]. Traditionally, the observations from the various sessions were administered using linking tables, where participants' names were assigned unique ids (running numbers). These ids are used to link observations from different sessions. Each individual datafile can be stripped of any information that reveals the identity of the participant. However, the existence of a linking table poses a risk to the participant's privacy if it is leaked. To overcome such problems, numerous procedures have been proposed including self-generated ids [47] and machine generated ids [2]. It is a goal to keep sessions short, and the process of establishing self-generated ids may divert the attention away from the experiment. Machine-based coding schemes [2] usually generate long and incomprehensible ids that do not contribute to participants' trust and perceived privacy. Long ids may seem like overkill in HCI experiments which often have around 12 participants [21]. There is also an ongoing debate about the ability of such methods to preserve participants' anonymity [30]. It has been found that participants do not always trust anonymization procedures [1]. Participants' decision about whether to give informed consent is balanced around the social sensitivity of the research questions and the common good that may result from the particular research [1].

Many research institutions have tightened their policies with the introduction of the General Data Protection Regulations (GDPR) to protect the privacy of individuals. Researchers need to apply for permission to store research data with information that identify participants such as their name, phone number, e-mail address and IP-addresses. Applying for such permissions can be time-consuming, bureaucratic, and daunting for students who are conducting their first HCI experiment. Students may be particularly discouraged from conducting longitudinal experiments as courses typically run across a limited period, making it impractical to collect formal permissions. Consequently, researchers may decide to settle for single session experiments to avoid the administrative burden of acquiring permissions. Decisions to omit the collection of data across multiple sessions may therefore impact the quality of the results negatively for certain types of research questions.

A simple procedure is therefore proposed for generating short human-readable linking ids to help researchers and students easily administer multi-session experiments while preserving the privacy of the participants. The procedure assumes that the experimenter initially has a list of names of potential participants. This list is used to search for an experiment-specific salt that gives minimum-length ids that uniquely discriminate the participants. During the experiment, the session is labelled with the unique id generated using the participant's name, the salt, and id length. The anonymity of the id is ensured by the principle of *k*-anonymity [39] where the short ids give multiple hits when applied to unrelated names. For example, with three-digit ids there will be statistically 1,000 hits per id with a brute-force attack using a list of one million names.

2 RELATED WORK

Longitudinal studies are sometimes used in HCI as it is often relevant to observe user experience over time [19, 31, 46]. Several frameworks for longitudinal observations within the field of HCI [16, 45] have not explicitly addressed anonymous linking of data.

However, the problem of linking of anonymous data from several sources has received a vast amount of attention within the medical domain, among researchers working on register data/microdata, and more recently among data scientists [12, 41, 43]. It has even been claimed that this problem is associated with most publications within computer science [23]. Especially fault tolerant linking has received much interest as records may be incomplete and contain errors. Incorrect record linkage in research may result in biased results [13].

Early approaches simply converted the names of the individuals using the soundex phonetic coding function [14, 29]. Linking was performed by comparing the phonetic codes. The phonetic coding step made the procedure tolerant to certain types of input errors such as spelling mistakes. The soundex function is lossy and irreversible, and originally it was considered to provide sufficient privacy [29]. The anonymity of soundex codes is probably contested by most today as these codes provide enough information to identify individuals in given contexts. A more recent soundex proposal added fake records to obfuscate the data [18].

Thoben et al. [40] explored anonymous record linkage using control numbers generated from one-way (lossy) functions. They investigated six control number coding schemes including (a) surname, date of birth, and sex, (b) first character of the surname, first name, birthdate, and sex, (c) soundex coded names, (d) sum of the ASCII codes of characters in the surname and family name string, date of birth, and sex, (e) three first characters of the surname, first name, month and year of birth, and sex, and (f) a variation on (e) without the month of birth.

Later approaches applied irreversible hash functions to the resulting phonetic representations [2, 3, 33]. Records were then linked by comparing hash codes. Simply encrypting records is not considered adequate [3] as encryption keys may get lost or be subject to cryptographic attacks. Although hash functions are irreversible, it is possible to conduct dictionary attacks, i.e., if we encode the list of all names from a phone book, the hash code would provide us with a reliable confirmation that a given person is in a dataset. Hash codes are therefore often salted, i.e., some string is added to the value before hashing [2]. Linking schemes based on hashing without phonetic coding have also been proposed [17]. Weber et al. [48] first reduced the information content in the data by using the two first characters from each of the first name and surname and the date of birth when creating the hash codes.

Phonetic methods have been criticized for proving more false positives than other methods [4, 9, 34, 44]. Most of the research effort during the last decade has therefore been on various probabilistic record linking procedures based on Bloom filters [2, 37]. In short, the Bloom filter approach involves extracting consecutive character pairs of the names into bigrams, then these bigrams are converted using several different hashing functions and mapped into a binary vector (typically 1,000 bits). Record linking is performed by comparing the bit vectors. It is thus possible to get partial matches, and the method is robust to many types of input errors.

Bloom filter methods have been criticized for being vulnerable to cryptanalysis attacks [5, 8, 22, 36, 37] and may not necessarily provide sufficient privacy to participants. One suggested improvement is to apply salting to bigram encodings [30]. Bloom filter approaches are often mentioned in the context of large data integration and are known for being slow [24]. Performance can be improved by splitting the data into blocks that are processed separately [32].

Hash codes are usually long. The well-known SHA and MD hashing functions with 160 and 128 bits respectively have been used to link data [2] and several Bloom filter proposals employ 1000 bits. If representing such hash codes in human readable format a 1,000-bit, Bloom filter coding requires a string of 250 hex digits, a 160-bit SHA hash requires 40 hex digits, and a 128-bit MD hash requires 32 hex digits (each hex-digit represents a group of four bits). Many of the algorithms that rely on long hash codes, such as Bloom filters, are intended for machine linking of large public records with information about millions of individuals. We argue that such ids are less suitable for manual administration of smaller experiments as strings of 32 hex digits (or more) would typically appear incomprehensible to participants. Clearly, such long strings of seemingly random sequences exceed the capacity of short-term memory (7+/-2) [28].

Self-generated identification ids constitute a manual approach to anonymous record linking procedures. In this approach each participant first answers a brief questionnaire. The questionnaire responses are used to generate unique ids that are used to tag the data from the sessions. Each time the participant attends a session the same self-generated id is created by the participant, allowing comparisons to be made across multiple sessions. Yet, the ids are intended to ensure the anonymity of participants. Many self-generated coding schemes have been proposed [6, 7, 11, 20, 42]. Yurek et al. [47] suggested a six-character id built from the answers to four questions: (a) the first letter of the mother's first name, (b) the number of older brothers (two digits), (c) month of birth (two digits), and (d) first name of middle name. Another example is Lippe et al.'s [25] id comprising (a) age, (b) first letters of middle name, the mother's first name, father's first name, grandmother's first name, own surname, number of older brothers, sisters, mother's siblings, modulus 13 of the first character of the first name, month of birth, and first letter of birthplace.

Self-generated ids may contain errors and Levenshtein distances have been explored as one remedy [38]. It has been found that self-generated ids with sensitive data can be vulnerable to attacks if the data are not sufficiently diverse [26, 27]. Glanti et al. [10] studied errors associated with self-generated ids and concluded that self-generated ids are inefficient for longitudinal studies.

3 THE HIDE PROCEDURE

In a typical multi-session experiment, one may use a linking table to analyze data across different sessions. A linking table typically assigns running numbers to the participants. Such running numbers do not reveal any personal information about the participants, but if the linking table is leaked the participants are no longer anonymous. The objective is thus to eliminate the linking table to protect the anonymity of the participants. An algorithm is needed that can assign such running (as close as possible to consecutive) numbers without knowledge about the list of participants.

Formally the problem can be defined as assigning N unique ids to N participants from a superset of M unique ids. The goal is to minimize M, and in the optimal case $N = M$. The effectiveness of the resulting ids can be defined as $E = N/M$, where an encoding is optimal if $E = 1$.

A novel aspect of the proposed approach involves an initialization step where the full list of names is used to search for a salt that minimizes the id lengths. The list of names is then discarded. To retrieve the linking id for a participant, the given salt is provided to the encoding procedure together with the participant's name.

The encoding comprises four steps: sorting the parts of each name alphabetically, converting the name to a phonetic representation, hashing, and truncating. The phonetic coding step and the hashing step are similar to those already used in several approaches [2, 3, 33] while the sorting step and the truncation step are unique to the HIDE procedure. The purpose of sorting the name parts alphabetically is to ensure that the procedure is robust to the name input order. For example, "Norman Donald" should give the same id as "Donald Norman".

Next, each part of the sorted name is converted to a phonetic representation using a modified soundex algorithm. The soundex algorithm preserves the first letter of a word, removes all vowels including the quasi-vowel *w*. The remaining consonants are mapped to a more course-grained phonetic representation using digit codes, namely 1: [b, f, p, v], 2: [c, g, j, k, q, s, x, z], 3: [d, t], 4: [l], 5: [m, n] and 6: [r]. Repeated digits are removed if they appear consecutively in the original name. For example, "Donald Norman" would be coded as "D543 N655". Unlike the original soundex algorithm which only retained the three first digits, the HIDE retains all the digits.

This phonetic simplification is intended to serve two purposes. First, it allows for more robust input with mistakes as it ignores incorrect vowels and is tolerant to incorrect transcription, for example, mistaking a *t* for a *d* and incorrect spelling of single/double letters. Second, the phonetic simplification may reduce the information content in the name, thereby leading to shorter ids.

The main purpose of the hashing step is to map the names uniformly across the set of possible ids. Moreover, the one-way hash function obfuscates the identities. The implementation reported herein relies on a simple integer (object) hash function built into the java programming language. Note that a salt is added to the phonetic representation prior to hashing.

The final step involves truncating the hash code by retaining the *d* least significant digits. For example, the hash code "57836278" would give an id of 278 if the ids are to comprise *d* = 3 digits. The purpose of the truncation step is to ensure anonymity. A hash code can be used to uniquely identify a participant, while a truncated hash code is not unique to a specific participant. Short ids provide a higher degree of anonymity than longer ids.

Clearly, there is a risk that two different names may end up with the same id. The procedure therefore involves an initialization step applied to the full list of participants. First, the list of names is checked for duplicates; next. the list of names in soundex representation is checked for duplicates to ensure that two names do not result in identical phonetic representations. Then, an exhaustive search is performed for a salt that leads to the fewest number of digits that uniquely discriminates all the names. This salt is added to each phonetic representation of the names before hashing. The 3,000 most frequent English words were used as potential salts. It is assumed that a simple word would be easier to memorize or transcribe than a salt comprising an arbitrary sequence of characters.

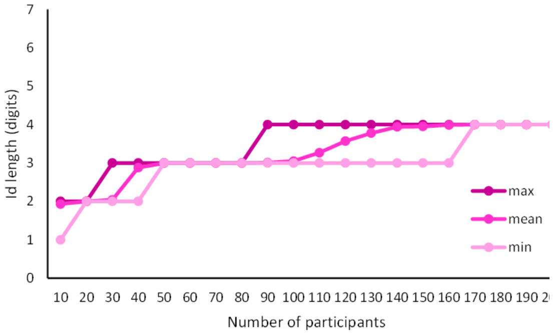

Figure 1: Id lengths of the HIDE procedure as a function of participant sample size in the best case, mean case, and worst case.

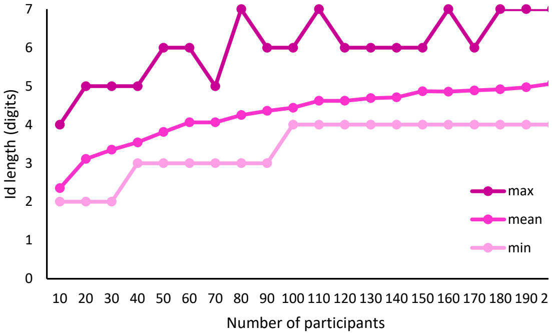

Figure 2: ID lengths without optimized salt as a function of participant sample size in the best case, mean case, and worst case.

4 EXPERIMENTAL EVALUATION

To demonstrate the viability of the proposed linking procedure, a list of 105,026 researchers worldwide based on Scopus data from a study by Ioannidis et al. [15] was used. The author list was cleaned as follows. First, names leading to 1,551 duplicate entries were identified and removed giving a list of 103,474 entries. Next, 5,911 names leading to duplicate phonetic representations were removed. The resulting list comprised 97,573 unique names. The lists are provided here [anonymized link] for reference.

First, the performance of the procedure was assessed for sample sizes of 10 to 200 participants in steps of 10. For each sample size the specified number of names were randomly drawn from the list of 97,573 names and input to the procedure. This step was repeated 100 times for each participant sample size to get representative results across a range of randomly drawn names. The minimum, maximum, and mean number of id digits obtained are plotted against participant sample size in Figures 1 and 2.

Figure 1 shows the results obtained using the HIDE procedure. The HIDE procedure successfully encoded 200 participants using 4 digits (E = 0.2) in the worst cases, 80 participants with 3 digits (E = 0.08) in the worst cases, and 20 participants with 2 digits (E = 0.2). In the best case 10 participants were encoded with 1 digit (E = 1), up to 40 participants with 2 digits (E = 0.4), and up to 160 participants with 3 digits (E = 0.16). The ability to anonymously encode 200 participants is sufficient for many types of HCI experiments which often have as little as 12 participants [21].

The results obtained without the phonetic coding step (direct hashing of names) are nearly identical to those presented in Figure 1 (and therefore not included). The sum of differences shows that HIDE only exhibited a negligible 0.03 overall improvement over the direct hashing of the names. Clearly, the phonetic coding step did not lead to the expected improved coding efficiency in the resulting hashes.

Figure 2 shows the results obtained when applying the procedure without the optimized salt. Clearly, the ids are longer when the optimized salt is not used. In fact, in several instances the worst-case yields ids with as much as 7 digits (E < 0.00001). With 7 digits privacy of participants probably cannot be preserved. In the mean case 4 digits are needed with 60 participants (E = 0.006), and 3 digits are needed with 30 participants (E = 0.03). These results demonstrate the importance of the optimized salt to effectively map the participants into a smaller range of id values.

Table 1: Coding capacity for larger participant sample sizes.

Participant sample size	Id lengths (digits)
400	5
800	5
1,600	6
3,200	6
6,400	7
12,800	20

Table 2: Effects of the 3000 salts based on a list of 163 names.

digit length	frequency
4	1,003
5	1,402
6	570
7	22

The detailed evaluation was limited to 200 participants due to the computational effort required. However, snapshots of how the procedure performs with larger experiments are shown in Table 1. With 5 digits it was possible to encode 800 participants. In practice, 5 digits are probably the largest number of digits that should be used as it can map 100,000 unique entries. With 6 digits it is one million possible mappings, which means that there is a larger risk being able to confirm that someone is a participant in an experiment. Table 1 shows that the coding scheme collapsed with 12,800 participants.

Figure 3 confirms that the optimized salts ensured shorter ids. To assess the effects of the salt, the results of a search for the optimal salt with 163 names were recorded as an example case (see Table 2). Most salts resulted in 5-digit ids, while about one third of the salts gave the minimum id length of 4 digits. Several salts resulted in 5, 6, and even 7-digit ids.

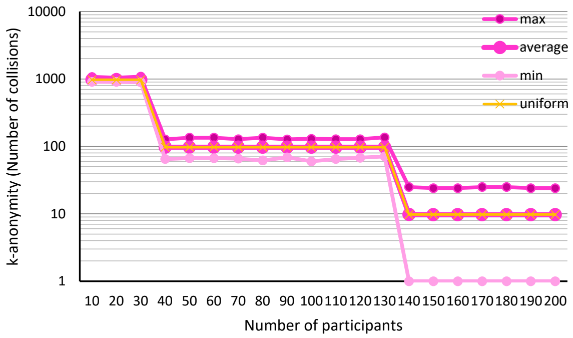

Figure 3: A log-linear plot of *k*-anonymity as a function of as a function of participant sample size in the best case, average case, worst case and uniformly distributed (expected). The data is based on the list 97,573 names.

To illustrate the anonymity provided by the proposed short ids, the full list of 97,573 names was fed back to the coding for one case from each sample size from 10 to 200. Figure 3 shows the distribution of names that maps to each of the possible id (both valid and invalid ids). With two-digit ids there were approximately 1,000 names from the list that mapped to each of the entries. With three-digit ids there were about 100 names mapped to each id, and with four-digit ids there were a mean of 10 names mapped to each id with a minimum of one name per id. Clearly, the scale of the statistics plotted in Figure 3 is connected to the size of this list. The size of this example is comparable to the population of Antigua and Barbuda (97,929 people in 2021).

5 DISCUSSION

With the list of more than 100,000 author names there was a 5.7% chance of phonetic collisions. With small experiments, which are the most common within HCI [21], the chance of phonetic collisions is low. One strategy for handling sound collisions such as “Lena Hansen”, “Lene Hanson”, and “Line Hansson” could be for the researcher to add an extra description using some mnemonic aid to help discriminate between the names. Examples include descriptions that give meaning and associations to the researcher when interacting with the participant “Lena Hansen decaffeinated”, “Lene Hanson parking permit”, and “Line Hansson administrator”. The experimenter would have to remember these special cases. Alternatively, the phonetic coding step can be omitted if there are collisions in the phonetic representation.

Obviously, there is a theoretical risk of hash collisions. However, the experimental evaluations showed that the salt prevents collisions in practice. The HIDE works with languages transcribed using the Latin alphabet. For other languages, such as Chinese, other schemes are needed in place of soundex. Alternatively, the phonetic coding step may be omitted by hashing the names directly, thereby sacrificing the tolerance to input errors.

Sorting the name parts alphabetically before encoding ensures that the parts of a name can be input in an arbitrary order. If one part of the name is missing (e.g., middle name or initial), the ids will not match, and the procedure will fail. Methods based on bigrams (including Bloom filters) are more tolerant to missing name parts.

The examples outlined herein use the participants’ full names. Alternatively, one may also use the participants’ first names or nicknames if these are unique. Small HCI experiments with less than 100 participants will typically yield ids with 2-3 digits. Most people are familiar with three-digit numbers, and these may therefore be perceived as less mystical and more trustworthy than long hash codes.

A JavaScript implementation of the procedure is made available through a simple web-interface running locally in the browser at [anonymized URL] as well as a java implementation that can be used for further experimentation at [anonymized URL].

6 CONCLUSION

The HIDE procedure for assigning short human-readable, anonymous, and error-tolerant linking ids to participants was presented. Experiments demonstrated the practicality of the approach. An implementation of the coding procedure running in the browser is available. It is hoped that the procedure will make it easier to gain participants' trust in the preservation of their anonymity. In addition, it is hoped that the procedure will encourage more HCI students and researchers to observe users over time.

REFERENCES

[1] Suzanne Audrey, Lindsey Brown, Rona Campbell, Andy Boyd, and John Macleod. 2019. Young people's views about consenting to data linkage: findings from the PEARL qualitative study. BMC medical research methodology 16, no. 1: 34.

[2] A. M. Benhamiche and J. Faivre. 1998. Automatic Record Hash Coding and Linkage for Epidemiological. Meth Inform Med 37: 271-7.

[3] Hocine Bouzelat, Catherine Quantin, and Liliane Dusserre. 1996. Extraction and anonymity protocol of medical file. In Proceedings of the AMIA Annual Fall Symposium, p. 323. American Medical Informatics Association.

[4] Rafael Camps and Jordi Daudé. 2003. Improving the efficacy of approximate searching by personal-name. Natural language processing and information systems.

[5] Peter Christen, Rainer Schnell, Dinusha Vatsalan, and Thilina Ranbaduge. 2017. Efficient cryptanalysis of bloom filters for privacy-preserving record linkage. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 628-640. Springer, Cham.

[6] S. Damrosch. 1986. Ensuring anonymity by use of subject-generated identification codes. Research in Health & Nursing 9: 61-63.

[7] Colleen DiIorio, Johanna E. Soet, Deborah Van Marter, Tammy M. Woodring, and William N. Dudley. 2000. An evaluation of a self-generated identification code. Research in nursing & health 23, no. 2: 167-174.

[8] Elizabeth A. Durham, Murat Kantarcioglu, Yuan Xue, Csaba Toth, Mehmet Kuzu, and Bradley Malin. 2013. Composite bloom filters for secure record linkage. IEEE transactions on knowledge and data engineering 26, no. 12 (2013): 2956-2968.

[9] Carol Friedman and Robert Sideli. 1992. Tolerating spelling errors during patient validation. Computers and Biomedical Research 25, no. 5: 486-509.

[10] M. Rosaria Galanti, Roberta Siliquini, Luca Cuomo, Juan Carlos Melero, Massimiliano Panella, Fabrizio Faggiano, and EU-Dap Study Group. 2007. Testing anonymous link procedures for follow-up of adolescents in a school-based trial: the EU-DAP pilot study. Preventive medicine 44, no. 2: 174-177.

[11] J. Grube, M. Morgan, and K. Kearney. 1989. Using self-generated identification codes to match questionnaires in panel studies of adolescent substance abuse. Addictive Behaviors 14: 159-71.

[12] Rob Hall and Stephen E. Fienberg. 2010. Privacy-preserving record linkage. International conference on privacy in statistical databases. Springer, Berlin, Heidelberg.

[13] Katie Harron, Chris Dibben, James Boyd, Anders Hjern, Mahmoud Azimaee, Mauricio L. Barreto, and Harvey Goldstein. 2017. Challenges in administrative data linkage for research. Big data & society 4, no. 2: 2053951717745678.

[14] David Holmes and M. Catherine McCabe. 2002. Improving precision and recall for soundex retrieval. In Proceedings. International Conference on Information Technology: Coding and Computing, pp. 22-26. IEEE.

[15] John PA Ioannidis, Jeroen Baas, Richard Klavans, and Kevin W. Boyack. 2019. A standardized citation metrics author database annotated for scientific field." PLoS biology 17, no. 8: e3000384.

[16] Jhilmil Jain and Susan Boyce. 2012. Case study: longitudinal comparative analysis for analyzing user behavior. In CHI '12 Extended Abstracts on Human Factors in Computing Systems (CHI EA '12). Association for Computing Machinery, New York, NY, USA, 793–800. DOI:https://doi.org/10.1145/2212776.2212852

[17] Stephen B. Johnson, Glen Whitney, Matthew McAuliffe, Hailong Wang, Evan McCreedy, Leon Rozenblit, and Clark C. Evans. 2010. Using global unique identifiers to link autism collections. Journal of the American Medical Informatics Association 17, no. 6: 689-695.

[18] Alexandros Karakasidis, Vassilios S. Verykios, and Peter Christen. 2011. Fake injection strategies for private phonetic matching. In Data Privacy Management and Autonomous Spontaneus Security, pp. 9-24. Springer, Berlin, Heidelberg.

[19] Evangelos Karapanos, John Zimmerman, Jodi Forlizzi, and Jean-Bernard Martens. 2009. User experience over time: an initial framework. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09). Association for Computing Machinery, New York, NY, USA, 729–738. DOI:https://doi.org/10.1145/1518701.1518814

[20] Kathleen A. Kearney, Ronald H. Hopkins, Armand L. Mauss, and Ralph A. Weisheit. 1984. Self-generated identification codes for anonymous

collection of longitudinal questionnaire data. Public Opinion Quarterly 48, no. 1B: 370-378.

[21] Kelly Caine. 2016. Local Standards for Sample Size at CHI. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16). Association for Computing Machinery, New York, NY, USA, 981–992. DOI:https://doi.org/10.1145/2858036.2858498

[22] Martin Kroll and Simone Steinmetzer. 2014. Automated cryptanalysis of bloom filter encryptions of health records. German Record Linkage Center, Working Paper Series, No. WP-GRLC-2014-05.

[23] Hye-Chung Kum, Ashok Krishnamurthy, Ashwin Machanavajjhala, Michael K. Reiter, and Stanley Ahalt. 2014. Privacy preserving interactive record linkage (PPIRL). Journal of the American Medical Informatics Association 21, no. 2: 212-220.

[24] Mehmet Kuzu, Murat Kantarcioglu, Ali Inan, Elisa Bertino, Elizabeth Durham, and Bradley Malin. 2013. Efficient privacy-aware record integration. In Proceedings of the 16th International Conference on Extending Database Technology, pp. 167-178.

[25] Megan Lippe, Bailey Johnson, and Patricia Carter. 2019. Protecting student anonymity in research using a subject-generated identification code. Journal of Professional Nursing 35, no. 2: 120-123.

[26] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. 2007. l-diversity: Privacy beyond k-anonymity. ACM Transactions on Knowledge Discovery from Data (TKDD) 1, no. 1: 3-es.

[27] Joe McGloin, Sherry Holcomb, and Deborah S. Main. 1996. Matching anonymous pre-posttests using subject-generated information. Evaluation review 20, no. 6: 724-736.

[28] George A. Miller, 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological review, 63(2), p.81.

[29] J. Y. Mortimer and J. A. Salathiel. 1995. 'Soundex'codes of surnames provide confidentiality and accuracy in a national HIV database. Communicable disease report. CDR review 5, no. 12: R183.

[30] Frank Niedermeyer, Simone Steinmetzer, Martin Kroll, and Rainer Schnell. 2014. Cryptanalysis of basic bloom filters used for privacy preserving record linkage. German Record Linkage Center, Working Paper Series, No. WP-GRLC-2014-04.

[31] David G. Novick, Baltazar Santaella, Aaron Cervantes, and Carlos Andrade. 2012. Short-term methodology for long-term usability. In Proceedings of the 30th ACM international conference on Design of communication, pp. 205-212.

[32] Guillermo Navarro-Arribas and Vicenç Torra. 2012. Information fusion in data privacy: A survey. Information Fusion 13, no. 4: 235-244.

[33] C. Quantin, C. Binquet, F. A. Allaert, B. Cornet, R. Pattisina, G. Leteuff, C. Ferdynus, and J. B. Gouyon. 2005. Decision analysis for the assessment of a record linkage procedure. Methods of Information in Medicine 44, no. 01: 72-79..

[34] Heather J. Rogers, and Peter Willett. 1991. Searching for historical word forms in text databases using spelling-correction methods: Reverse error and phonetic coding methods. Journal of Documentation.

[35] I. Scott MacKenzie and Shawn X. Zhang. 1999. The design and evaluation of a high-performance soft keyboard. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '99). Association for Computing Machinery, New York, NY, USA, 25–31. DOI:https://doi.org/10.1145/302979.302983

[36] Sean M. Randall, Anna M. Ferrante, James H. Boyd, Jacqueline K. Bauer, and James B. Semmens. 2014. Privacy-preserving record linkage on large real world datasets. Journal of biomedical informatics 50: 205-212.

[37] Rainer Schnell, Tobias Bachteler, and Jörg Reiher. 2009. Privacy-preserving record linkage using Bloom filters. BMC medical informatics and decision making 9.1: 41.

[38] Rainer Schnell, Tobias Bachteler, and Jörg Reiher. 2010. Improving the use of self-generated identification codes. Evaluation Review 34, no. 5: 391-418.

[39] Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10, no. 05: 557-570.

[40] W. Thoben, H-J. Appelrath, and S. Sauer. 1996. Record linkage of anonymous data by control numbers. In From Data to Knowledge, pp. 412-419. Springer, Berlin, Heidelberg, 1996.

[41] Stanley Trepetin. 2008. Privacy-preserving string comparisons in record linkage systems: a review. Information Security Journal: A Global Perspective 17, no. 5-6: 253-266.

[42] Jaroslav Vacek, Hana Vonkova, and Roman Gabrhelík. 2017. A successful strategy for linking anonymous data from students' and parents' questionnaires using self-generated identification codes. Prevention Science 18, no. 4: 450-458.

[43] Dinusha Vatsalan, Peter Christen, and Vassilios S. Verykios. 2013. A taxonomy of privacy-preserving record linkage techniques. Information Systems 38.6: 946-969.

[44] Dinusha Vatsalan, and Peter Christen. 2012. An iterative two-party protocol for scalable privacy-preserving record linkage. In Proceedings of the Tenth Australasian Data Mining Conference-Volume 134, pp. 127-138.

[45] Misha Vaughan, Catherine Courage, Stephanie Rosenbaum, Jhilmil Jain, Monty Hammontree, Russell Beale, and Dan Welsh. 2008. Longitudinal usability data collection: art versus science? In CHI '08 Extended Abstracts on Human Factors in Computing Systems (CHI EA '08). Association for Computing Machinery, New York, NY, USA, 2261–2264. DOI:https://doi.org/10.1145/1358628.1358664

[46] Jorick Vissers, Lode De Bot, and Bieke Zaman. 2013. MemoLine: evaluating long-term UX with children. In Proceedings of the 12th International Conference on Interaction Design and Children (IDC '13). Association for Computing Machinery, New York, NY, USA, 285–288. DOI:https://doi.org/10.1145/2485760.2485836

[47] Leo A. Yurek, Joseph Vasey, and Donna Sullivan Havens. 2008. The use of self-generated identification codes in longitudinal research. Evaluation review 32, no. 5: 435-452.

[48] Susan C. Weber, Henry Lowe, Amar Das, and Todd Ferris. 2012. A simple heuristic for blindfolded record linkage. Journal of the American Medical Informatics Association 19, no. e1: e157-e161.