DEMONSTRATION OF A PRIVACY-PRESERVING SYSTEM THAT PERFORMS AN UNDUPLICATED ACCOUNTING OF SERVICES ACROSS HOMELESS PROGRAMS

by

Latanya Sweeney, PhD

Associate Professor of Computer Science, Technology and Policy
Director, Data Privacy Lab
School of Computer Science
Carnegie Mellon University, Pittsburgh, PA
latanya@cs.cmu.edu

FINAL REPORT

U.S. Government Release

October 2008

Table of Contents

INTRODUCTION

BACKGROUND

METHODS

RESULTS

DISCUSSION

Index of Figures

1. Executive Summary

Over the last two years, the United States Department of Housing and Urban Development ("HUD") reviewed ways to perform a national unduplicated accounting of visit patterns across homeless programs, while respecting the confidentiality of those clients who visit domestic violence homeless shelters.

The goal of the work reported in this writing was to demonstrate a system that performs an accurate unduplicated accounting across homeless programs with guarantees of privacy protection for clients of domestic violence homeless shelters.

HUD sponsors locally administered Homeless Management Information Systems ("HMIS") in order to collect data needed for an annual report HUD provides to Congress termed the "Annual Homeless Assessment Report (AHAR)" [1]. A HMIS is a computerized data collection and processing system designed to capture person-specific information over time from homeless persons being serviced by any homeless program, including domestic violence homeless shelters. Information gathered from all homeless service programs that are geographically co-located is compiled by a HMIS operated by a "Planning Office" (called a "Continuum of Care" or "CoC" in HUD documents) that is local to those programs. Information collected at homeless programs is not directly forwarded to HUD. Instead, the local Planning Office de-duplicates and forwards de-identified, unduplicated aggregate information to HUD.

Special privacy considerations are given to the clients of domestic violence homeless shelters so that client information provided by a domestic violence homeless shelter to a HMIS cannot be re-identified to the clients who are the subjects of the shared information. HMIS are to gather information from local domestic violence homeless shelters in such a way that client confidentiality is maintained yet an accurate unduplicated accounting of visit patterns can still be achieved across homeless programs by planning offices.

In initial steps to protect privacy, HUD modified the fields of information it recommends domestic violence homeless shelters share with a HMIS [1]. The fields HUD recommends are termed the "Universal Data Elements." Rather than using client names or Social Security numbers in the Universal Data Elements, HUD introduced the notion of assigning a unique identifier ("UID") to clients of domestic violence shelters [2].

This paper reports on the use of a technology ("PrivaMix") for constructing UIDs and performing de-duplication such that an accurate unduplicated accounting results while protecting the privacy of the clients who are the subjects of the UIDs (Section 8).

In a real-time experiment, a "PrivaMix Demonstration System" computed an accurate unduplicated accounting using real-world data from homeless programs in Des Moines, Iowa ("the Iowa Experiment"). This writing examines the experiment (Section 10), the data elements shared, the client information used to construct UIDs , the algorithms that generated those results (Section 9), and the privacy implications of results (Section 11).

Here is a summary of performance findings.

The PrivaMix Demonstration System introduced no errors in the unduplicated accounting. It performed exactly as if plain text was used even though Client information was provably never shared with the Planning Office or the other shelters. Section 9 introduces the PrivaMix Demonstration System. Section 10 reports results from the Iowa Experiment.

The client's combination of {*date of birth*, *first three letters of first name*} were used to generate secure UIDs. This writing terms this the "Privacert encoding" as Privacert first proposed its use. Experiments compared Privacert's proposed method with using Social Security numbers, and two methods currently in use by Servicepoint[1]. Privacert's method encountered fewer fields having omissions or errors than the other methods, and used fields in which clients provided more consistent values than the fields used by the other methods. In performing an unduplicated accounting, the Privacert method proved more accurate than the other approaches. Section 10 reports on a comparison of the use of demographics in forming UIDs. (While Privacert proposed this encoding, it is important to note that the PrivaMix System is not specific to any particular encoding method.)

Modifications to the shared data elements improved privacy without loss of reporting ability. Participants in the Iowa Experiment shared *year of birth* with the Planning Office instead of the full month, day, and year of birth as currently recommended in the Universal Data Elements. Doing so, reduced the likelihood of re-identification using publicly available data from 87% to 0.04% (see Section 4.5). While this is an important improvement, other privacy threats remain in the data elements (see Section 11) and are further discussed below.

PrivaMix guarantees privacy protection for UID creation and use in de-duplicating. As noted above, these privacy protections had no adverse effect on de-duplication. However, privacy threats related to the selection of which client-level data elements to associate with UIDs remains. These problems reside beyond the scope of the PrivaMix Demonstration System (or any other UID technology). Below is a discussion of these vulnerabilities and a description of how post-processing anonymization can be added to the PrivaMix System to remedy them.

Data linkage vulnerabilities exist when a Planning Office subsequently shares de-duplicated results with the HMIS (Section 8). Collusion between the HMIS and Planning Office can reveal the identifies of clients. In environments where collusion is possible between the Planning Office and the HMIS, additional safeguards are necessary to combat this threat (Section 12).

The Iowa Experiment posed a situation in which the community would likely rely on HMIS staff to perform the functions of the Planning Office, thereby introducing privacy risks due to possible collusion.

One problem is data linkage on demographics appearing in the shared client-level data. Section 11 reports that 36% of the Iowa clients had uniquely occurring combinations of {*year of birth*, *gender*, *5-digit ZIP*}, and the number jumps to 55% when including {*race*, *ethnicity*}.

[1] Servicepoint is a product of Bowman Systems, servicing more than 30,000 clients in 45 states. They are a national leader in providing HMIS services. For more information, see http://www.bowmansystems.com/products.html.

Of course, just because a client has uniquely occurring demographics does not mean she is identifiable. The party seeking to re-identify data (termed "the linker" in this writing) must hold sufficient information to exploit this uniqueness. Section 2 reports that the likelihood in the USA of unique re-identifications of clients based on {*year of birth*, *gender*, *5-digit ZIP*} is only 0.04%. If the linker only has access to publicly available data (e.g., a voter list), then the likelihood a re-identification using these demographics is 0.04%. On other hand, if the linker is the HMIS in Iowa, which often contains non-domestic violence service records related to the same clients, then the likelihood of a re-identification using these demographics is about 36%.

One remedy to help thwart unwanted linking by the HMIS using demographic data elements is to only share the most general version of the data elements that still enable production of the AHAR. Section 11 reports that {first 3 digits of ZIP, gender, AHAR age ranges} was unique for 6% of the Iowa clients and was 11% when including {race, ethnicity}. This is a dramatic improvement, and even though it is not the only solution needed, sharing only the most general values lowers privacy risks overall.

A second problem is re-identification due to the linker exploiting the exact entry and exit dates appearing in the data (Section 11). A somewhat effective remedy is to replace exact dates of service with number of days of service or with time periods (e.g., overnight, 2-14 days, 15-30 days, 30 plus days). Section 11 provides more detail.

In order to prevent collusion when the Planning Office and the HMIS consist of the same personnel, it is necessary to use additional privacy safeguards, beyond those implemented in the PrivaMix Demonstration System or made possible by changes described to the Universal Data Elements. These safeguards involve post de-duplication anonymization. After PrivaMix de-duplication completes, additional processing would occur before it releases results to the Planning Office. Possibilities for post processing include: replacing client-level results with pivot tables that show aggregate count information for combinations of data elements; replacing client-level data with an overall final report (e.g., the AHAR itself); or, suppressing and generalizing outliers in the client-level results. Each of these approaches can provide additional and sufficient privacy protection, by replacing client-specific results with appropriately generalized ones. Section 12 describes these in detail.

In comparison to other approaches, the PrivaMix approach does not require domestic violence homeless shelters to share identifiable client data with a third party, a trusted third party, or an HMIS directly, as would a reporting service or centralized data storage, and provides better performance than encoding, hashing, encryption, scan cards, biometrics, and consent at constructing privacy-preserving UIDs. Section 9.9 and Figure 1 provides a comparison.

In conclusion, the PrivaMix Demonstration System achieved an accurate unduplicated accounting in the Iowa Experiment, and with the additional post processing anonymization described above, can do so while maintaining client privacy even in an environment in which the Planning Office and the HMIS are the same people.

More detailed information and recommendations appear below. These recommendations concern information collected from clients of domestic violence homeless shelters (termed "Clients" and "Shelters") and are not necessarily intended to be more generally applied to other homeless populations whose information may be captured in a HMIS. Figure 2, in Section 1.9, provides a quick summary of all recommendations.

1.1 General recommendations

Recommendation #1: Coordination of privacy protection schemes is necessary across planning offices that service a geographical region in which shelters within the region report to different planning offices but service some of the same clients. Lack of coordination can distort the unduplicated accounting. (For more information, see Section 3.3.)

Recommendation #2: A Shelter may assign a unique person identification number (PIN) to internally identify a client, but it should not share the client's PIN externally. PINs that include the Client's name, Social Security number, or other characteristic may be used alone or in combination with other data elements to re-identify a Client. Any characteristic not allowed as a data element or a UID, should not be used as an externally shared PIN. (For more information, see Section 3.5.)

Recommendation #3: If a Planning Office produces a De-identified Dataset from the HMIS data collected from Shelters, the De-identified Dataset should not include any original Personal Identification Numbers (PINs), Unique Identification numbers (UIDs), or Household Identification numbers. (For more information, see Section 3.6.)

Recommendation #4: A Shelter should release Client information to the Planning Office some time after the Client has left the shelter. (For more information, see Section 4.1.)

Recommendation #5: Shelters and planning offices should train personnel on the responsibilities and accepted practices for collecting, storing and sharing client information. (For more information, see Section 4.1.)

Recommendation #6: Unique Identification numbers (UIDs) values assigned to Clients of domestic violence shelters by Shelters should not be used (i.e., stored or referenced) by any non-HMIS program to which the Clients may participate in order to limit unwanted linking. For more information, see Section 4.2.)

Recommendation #7: Shelters and Planning Offices are already required to issue and post privacy notices to clients about the data collection, sharing, and linking practices of the shelters and planning offices in which the client's data will be part [1]. Beyond the role this requirement plays as a Fair Information Practice, this requirement is also important to help ensure the integrity of the information a client provides in forming the client's UID. (For more information, see Section 4.2.)

Recommendation #8: The fields *date of birth* and *ZIP code of last residence*, which are among the data elements Shelters share with Planning Offices, should contain information less specific than the month, day, and year of birth and all 5 digits of the ZIP (or postal) code. (For more information, see Section 4.5 and Section 7.1.)

Recommendation #9: A Planning Office may generate a "De-identified Dataset" from collected Shelter data to compute the unduplicated accounting. If so, the Planning Office should only use the Universal Data Elements in computing the De-Identified Dataset and remove (or obscure) elements from the De-identified Dataset that may appear in other data held by the Planning Office to limit secondary linking to other data held by the Planning Office. (For more information, see Section 4.6.)

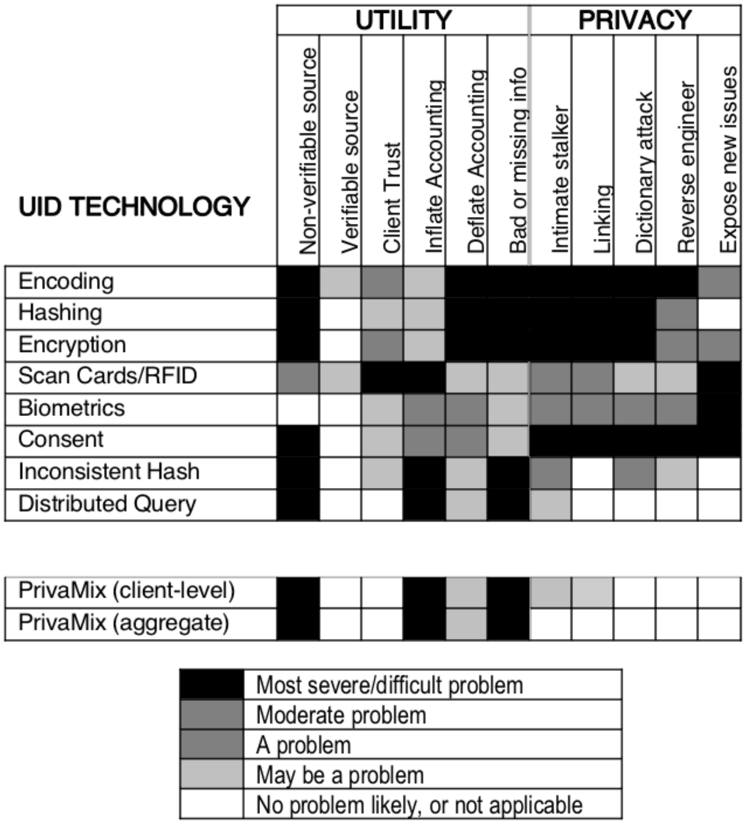
Recommendation #10: Personnel in the Planning Office should sign a data use agreement with Shelters or provide notice to Shelters that either disallows the linking of the De-Identified Dataset to any other data or makes explicit the linking intended. (For more information, see Section 4.6.)

Recommendation #11: Given a "Proposed Solution" (i.e., a UID technology bundled with policies and practices for the construction, maintenance and use of a UID technology for clients of domestic violence homeless shelters), a person skilled in statistical, computational and/or legal principles, as appropriate, should certify in writing that the Proposed Solution has a minimal risk of re-identification when the solution is considered with other publicly and readily available information and techniques. Such writing should address vulnerabilities for inappropriate re-identifications by various categories of insiders. This is termed a "compliance statement" and should be made available for inspection. (For more information, see Section 5.5.)

Recommendation #12: Given a Proposed Solution, a person skilled in statistical and/or computational principles, as appropriate, should certify in writing that the Proposed Solution provides a reasonably accurate unduplicated accounting of client visit patterns to shelters within the regional setting it is to be deployed. Such writing should include possible false match and missed match rates. This statement is termed a "warranty" and should be made available for inspection. (For more information, see Section 5.5.)

1.2 Recommendations regarding UID technologies

The following recommendations result from assessments performed on the initial UID technologies explored by Shelters and Planning Offices. The list of initial technologies appear in Figure 1.



UID TECHNOLOGY	UTILITY						PRIVACY				
	Non-verifiable source	Verifiable source	Client Trust	Inflate Accounting	Deflate Accounting	Bad or missing info	Intimate stalker	Linking	Dictionary attack	Reverse engineer	Expose new issues
Encoding	■	░	▩	░	■	■	■	■	■	■	▩
Hashing	■	□	░	░	■	■	■	■	■	▩	□
Encryption	■	□	▩	░	■	■	■	■	■	▩	▩
Scan Cards/RFID	▩	░	■	■	░	░	▩	▩	░	░	■
Biometrics	□	□	░	▩	▩	░	▩	▩	▩	▩	■
Consent	■	□	░	▩	▩	░	■	■	■	■	■
Inconsistent Hash	■	□	░	■	░	■	▩	□	▩	░	□
Distributed Query	■	□	□	■	░	■	░	□	□	□	□
PrivaMix (client-level)	■	□	□	■	░	■	░	░	□	□	□
PrivaMix (aggregate)	■	□	□	■	░	■	□	□	□	□	□

■	Most severe/difficult problem
▩	Moderate problem
▩	A problem
░	May be a problem
□	No problem likely, or not applicable

Figure 1. Technologies considered for UIDs. The top group are the initial technologies.

Recommendation #13: If the technology for constructing UIDs uses non-verifiable information from the client, then instruments that instill client trust in the overall system should be deployed; otherwise, the UID should use verifiable source input from clients. (For more information, see Section 6.9.)

Recommendation #14: If the technology for constructing UIDs involves encryption or hashing, then "strong" cryptographic methods should be used and the description of the method should be included in the warranty or compliance statement. (For more information, see Section 6.9.)

Recommendation #15: If the technology for constructing UIDs involves encryption or hashing, then accompanying practice should control access to and document an audit trail of specific uses of the encryption/hashing function. A description of these practices related to the capture and auditing of uses of the encryption/hashing function should be included in the warranty or compliance statement. (For more information, see Section 6.9.)

Recommendation #16: If the technology for constructing UIDs involves scan cards, then accompanying practices are needed to avoid issuing multiple cards to the same client and to prevent card sharing and swapping among clients. A description of practices related to avoiding these unwanted activities should be included in the warranty or compliance statement. (For more information, see Section 6.9.)

Recommendation #17: In cases where consistent UIDs are assigned to Clients over time, once Planning Offices link and de-duplicate Client visits, stored copies of the linked information should have all UIDs removed. (For more information, see Section 6.9.)

1.3 VAWA-based recommendations

In January 2006, Congress passed The Violence Against Women and Department of Justice Reauthorization Act of 2005, H.R. 3402 ("VAWA") which raised the privacy standard for UID technologies to guarantee clients cannot be re-identified. Recommendations related to the impact of VAWA on HMIS data elements and UID technologies appear below.

Recommendation #18: The fields *date of birth* and *ZIP code of last residence*, which are among the data elements in the Universal Data Elements, must contain information less specific than the month, day, and year of birth and all 5 digits of the ZIP (or postal) code in order to thwart linking. (For more information, see Section 4.5 and Section 7.1.) This is a strengthening of Recommendation #8.

Recommendation #19. The technology used to construct and de-duplicate UIDs must satisfy VAWA's requirements limiting re-identification. Consent and biometrics appear unable to satisfy the privacy standard established by VAWA. Encoding, hashing, and encryption may enable unwanted linking, and if so, pose grave concerns in attempts to use them to satisfy VAWA's privacy standard. Scan cards and RFID tags may be used, depending on the information appearing on (or within) the card. (For more information, see Section 7.2.)

1.4 PrivaMix recommendations

PrivaMix (Section 8) combines a form of inconsistent hashing (Section 6.7) with distributed query (Section 6.8) in three steps. These form the "PrivaMix Protocol."

The first step involves the assignment of UIDs. The same client gets different UIDs at different Shelters and can get the same UID at the same Shelter. This is done by using a strong one-way function having the commutative property. We term this a "PrivaMix function" (see Section 8.3 for requirements). Each Shelter computes a UID for a Client by applying the PrivaMix function to both a private value held by the Shelter and the source information provided by the Client (Section 8), thereby yielding different UIDs at different Shelters for the same Client.

In the second step, Shelters ship Client visit information to the Planning Office. Each record contains requested visit information and the Client's UID. At the end of this step, the Planning Office has visit details for all Clients at all Shelters, but does not know which UIDs relate to the same Clients across Shelters.

In the third step, Shelter and Planning Office machines communicate over a network to de-duplicate UIDs. We term the network of machines, a "PrivaMix Network." Each Shelter applies the PrivaMix function to its private value and the UIDs from all the other Shelters once in a process we term "mixing." After all Shelters finish mixing, complete mixes for UIDs will only be the same if the original Client source information was the same. This identifies which UIDs refer to the same Client.

There are variations to the generic PrivaMix Protocol to address particular issues.

PrivaMix Variation 1: Shelters mix among themselves, without the Planning Office (Section 8.2.2).

PrivaMix Variation 2: Shelters check that UIDs are legitimate (Section 8.2.3).

PrivaMix Variation 3: Matching UIDs to Universal Data Elements (Section 8.2.4).

PrivaMix Variation 4: Providing aggregate count distributions, not Client-level data (Section 8.2.5).

PrivaMix Variation 5: Anonymizing client-level data (Section 8.2.6).

PrivaMix Variation 6: Using web browsers for mixing (Section 8.2.7)

Recommendations below relate to using PrivaMix as a UID technology.

<u>Recommendation #20:</u> When using PrivaMix as a UID technology, care should be taken to avoid multiple Shelters from having the same private value. The Shelter's private value customizes the PrivaMix function to the Shelter. If multiple Shelters inadvertently have the same private value, then those Shelters assign exactly the same UIDs to the same clients. In most uses of PrivaMix, the UIDs will only be used for one-time mixing. In these cases, it is okay if Shelters inadvertently select the same private value though the likelihood of such should be rare. (For more information, see Section 8.1.)

<u>Recommendation #21:</u> When using PrivaMix as a UID technology, if the visit data is transmitted to the Planning Office over the PrivaMix network of Shelter and Planning Office machines, then appropriate computer security standards for the storage of Client information should be enforced because these machines contain Client source and visit information. (For more information, see Section 8.1.)

<u>Recommendation #22:</u> If desirable, use a variation of the PrivaMix Protocol to have a party other than the Planning Office orchestrate mixing. One variation (Section 8.2.2) describes how Shelters perform mixes among themselves and then forward de-duplicated results to the Planning Office. Another variation (Section 8.2.7) describes how a third-party might orchestrate de-duplication and then forward results to the Planning Office.

<u>Recommendation #23:</u> Thwarting data linkage threats requires further privacy consideration, realized as variations of PrivaMix and/or dictates on data elements. Rather than PrivaMix providing Client-level data to the Planning Office, PrivaMix can alternatively provide aggregate de-duplicated count distributions (Section 8.2.5). A way to help thwart data linkage threats within PrivaMix while still providing Client-level data is to anonymize the data after de-

duplication (Section 8.2.6). An alternative that lies outside of PrivaMix is to chose non-identificable Client-level data elements (Section 11).

Recommendation #24: An economical implementation of the PrivaMix Protocol involves using traditional web browsers already provided with computers (Section 8.2.7). Doing so has the advantage that no dedicated machine is needed, that no additional software has to be installed, and that no intense user training is needed.

Recommendation #25: A PrivaMix function (**F**) must satisfy the following six requirements (Section 8.3):

(1) Inconsistent assignment: different shelters should generate different initial mix values for the same clients.
(2) One-way function: **F** must be a one-way function.
(3) Commutative: **F** must be a commutative cipher.
(4) Privacy: the secret client information cannot be learned given the sharing of complete and sub-mixes.
(5) Collision-free: mixes from **F** must be collision-free.
(6) Correctness: all complete mixes for the same client must be the same. Complete mixes for different clients should not be the same.

Here are seven statements claimed about PrivaMix. These form the basis of the recommendations that follow them.

> Usability claim. Communication time is linear in the number of Shelters. (Section 8.4.1.)
>
> Correctness claim. If the complete mixes are the same, the Clients representing the original UIDs presented the same source information.. (Section 8.4.2.)
>
> Privacy claim. A dictionary attack by the Planning Office will not yield reliable re-identifications. (Section 8.4.3.)
>
> Privacy claim. Compromising a Shelter will not help the intimate stalker learn where a targeted Client is (or has been). Similarly, compromising the Planning Office will not help the intimate stalker learn where a targeted Client is (or has been). (Section 8.4.4.)
>
> Privacy claim. Even if the Planning Office pads the UIDs with known values, the Planning Office does not learn Client source information. (Section 8.4.5.)
>
> Limitation. If the Planning Office and at least one Shelter collude, the Planning Office can learn Client source information about the Shelter's Clients and the Shelter can learn other Shelters its Clients visited. (Section 8.4.6.)
>
> Limitation. If during the de-duplication protocol, the intimate stalker compromises both the Planning Office and a Shelter the targeted Client visited, the intimate stalker can learn the locations of all Shelters the Client visited. In addition, the Planning Office can learn the source information for that Client. (Section 8.4.7.)

Recommendation #26. Each Shelter must select a sufficiently private value so that efforts by the Planning Office to exhaustively compute all combinations of Shelter private values and Client source information (a dictionary attack) are not feasible. Most likely a Shelter's computer will be required to select a private value 512 bits or larger as appropriate and most likely randomly selected at the start of each reporting period. (For more information, see Section 8.4.)

Recommendation #27: To help thwart the possibility of the Planning Office or other Shelters learning a Shelter's private value, a Shelter may not even explicitly know its own private value for a reporting period –i.e., the computer program may generate it internally and not explicitly reveal it. (For more information, see Section 8.4.)

Recommendation #28: To help thwart the possibility of the Planning Office or other Shelters learning a Shelter's private value, a Shelter may make its private value available to its copy of the PrivaMix function only while mixing over the PrivaMix Network. Other parties should not be able to invoke a Shelter's PrivaMix function with the Shelter's private value. (For more information, see Section 8.4.)

Recommendation #29: In order to prevent the Planning Office from padding UIDs with known values, the original PrivaMix approach should be modified to validate the number of UIDs and/or to mix UIDs without Planning Office involvement. (See Variation 1 and Variation 2 in Section 8.2 for details and Section 8.4 for motivation.)

Recommendation #30: Care must be taken to combat possible collusion between the HMIS and the Planning Office because in many geographical regions, the staff of the HMIS is the same staff as the Planning Office (or CoC) and because there is a desire to de-duplicate visits across the domestic violence homeless shelters and the HMIS (not the domestic violence homeless shelters alone). As a participant in PrivaMix, a HMIS poses a significant threat to Client re-identifications because a HMIS will usually contain most (if not all) Clients who visit any domestic violence homeless shelter. Remedies include having PrivaMix provide only aggregate information or provably anonymizing released data elements. (See Section 12 for details and Section 8.4 for motivation.)

Recommendation #31: Client records Shelters provide to the Planning Office should only include Clients who are no longer residing at the Shelter. This is a helpful recommendation, but not wholly satisfactory because Clients may re-visit previously visited Shelters. (For more information, see Section 8.4.)

Recommendation #32: The Planning Office should destroy all copies of the original UIDs once the de-duplication is complete. Doing so, limits the opportunity for compromise. (For more information, see Section 8.4.)

Recommendation #33: A specific implementation of a system that uses the PrivaMix approach requires revisiting claims and limits specific to implementation details. Differences in implementations may include communication flow (e.g. Planning Office in the middle or Shelter-to-Shelter), information content (e.g., a stream of values, or a list of values with their originating Shelter), and selection of the privately held Shelter value (.e.g., random selection, or pre-selection). (For more information, see Section 8.4.)

In comparing PrivaMix with the UID technologies discussed earlier, PrivaMix performs comparable to inconsistent hashing (Section 6.7) and distributed query (Section 6.8) making it generally better than encoding (Section 6.1), hashing (Section 6.2), encryption (Section 6.3), scan cards and RFIDs (Section 6,4), biometrics (Section 6.5), and consent (Section 6.6) at protecting privacy. Yet, the utility of its de-duplicated results is better than encoding, hashing, encryption, scan cards and RFID, but not better than biometrics or consent. (For more information, see Section 8.5.)

1.5 The PrivaMix Demonstration System

In 2007, Privacert implemented a version of PrivaMix for a real-world experiment; we term this software the "PrivaMix Demonstration System." Here is a quick summary of its highlights.

- uses regular computers operating over the Internet
- each participant (Shelter and Planning Office) has its own machine
- data is shared using standard comma-delimited text files
- the Planning Office machine coordinates mixing
- final de-duplicated results don't include UIDs or complete mixes, just sequential numbers

Because there are numerous variations and many ways to implement the PrivaMix Protocol, Section 9 describes the details of the PrivaMix Demonstration System specifically. Section 10 explains its use in the real-world experiment. Below is a brief description of the PrivaMix Demonstration System.

In the PrivaMix Demonstration System, each participating machine runs special software devoted to this task. Shelter machines run one edition of the software program ("the Shelter Edition"). The Planning Office machine runs a different edition ("the CoC Edition"). These editions differ because the responsibilities of Shelters and the Planning Office in the PrivaMix protocol are different. (For more information, see Section 9.)

Operation of the PrivaMix Demonstration System is extremely simple. If Shelters and the Planning Office use default settings, then operation is as simple as loading the Client information and clicking one button. (For more information on user options and screen shots, see Appendix A.)

The PrivaMix Demonstration System has minimal machine requirements, which means almost any computer system sold today is sufficient for use. However, the machine must have access to the Internet. (For more information, see Section 9.1.)

The Shelter provides an initial comma-delimited text file for processing, which has the fields that comprise the Client's source information appearing as the leftmost fields. The remaining fields on the line are fields associated with the Client's visit to the Shelter, presumably the Universal Data Elements associated with that Client. After the Shelter machine computes UIDs for each Client from Client source information, it produces a comma-delimited file replacing the leftmost fields with Client UIDs. Shelter machines then transfer the resulting comma-delimited text file to the Planning Office as encrypted content over an Internet connection. (For more information, see Section 9.5.)

While the PrivaMix Demonstration System does not dictate which Client fields to use as source information, precautions are needed. Below are two important precautions. (For more information, see [32] and Section 9.4.)

1. Care must be taken that sufficient variability exists in the fields so that resulting UIDs have a sufficiently wide range of possible values.

2. Care must also be taken to make sure that different Clients are not likely to have to the same set of values appearing in the source information.

In the PrivaMix Demonstration System, the Planning Office orchestrates mixing as described in the generic PrivaMix Protocol (Section 8.2). The Planning Office sends values to each Shelter, one Shelter at a time, to mix, such that each Shelter mixes each UID once.

After mixing completes, the PrivaMix Demonstration System performs de-duplication on the Planning Office machine matching complete mixes across Shelter data. All values are held in the computer's memory. No information appears on the hard drive. (For more information, see Section 9.7.)

Before making final de-duplicated results available to the Planning Office, the PrivaMix Demonstration System removes all UIDs, replacing them with numbers from 1 to the total number of distinct Clients. The Planning Office does not receive a copy of the UIDs or complete mixes, only the results of de-duplication. (For more information, see Section 9.8.)

1.6 The Iowa experiment

In a real-time experiment with three shelters, an HMIS and a Planning Office, a “PrivaMix Demonstration System” computed an accurate unduplicated accounting using real-world data from homeless programs in Des Moines, Iowa (“the Iowa Experiment”). Here is a summary of experimental results. For details, see Section 10.

The experiment used laptops with wireless broadband network, with the software loaded and pre-configured for operation. Standardizing the machines allowed the experiments to focus efficiently and narrowly on performance.

Subjects were clients whose data appeared at participating shelters and the HMIS in a previous six-month time period. The actual subjects are not clients of domestic violence (“DV”) homeless shelters, but are clients of homeless family shelters (not domestic violence specific). Using non-DV shelters allowed us to compare computed de-identified results with results derived manually using fully identified data. Of course, the generalizability of these experiments assume there is no difference between DV and non-DV data collection.

A key component in de-duplicating UIDs is the Client source information used to construct the UIDs. Fields having omissions or errors can render UIDs useless. While the PrivaMix Demonstration System works with any Client source information, Privacert proposed to use the first three letters of the first name and the date of birth. Experiments compared Privacert's proposed method with using Social Security numbers, and two methods currently in use by

Servicepoint. Privacert's method encountered fewer fields having omissions or errors than the other methods, and used fields in which clients provided more consistent values than the fields used by the other methods. In performing an unduplicated accounting, the Privacert method had the lowest number of errors.

After constructing UIDs, shelters, the HMIS, and Planning Office conducted a real-time duplication using the laptops located at their facilities. The PrivaMix Demonstration System performed exactly as if plain text was used even though sensitive Client source information was provably never shared with the Planning Office or the other Shelters. No errors were introduced.

1.7 Changes to the Universal Data Elements

The generic PrivaMix approach solves privacy and utility problems related to the assignment and de-duplication of UIDs. However, privacy threats may remain from data linkage capabilities afforded by the Universal Data Elements. Below are recommendations related to demographics appearing in the Universal Data Elements.

Recommendation #34: The AHAR does not require the demographic specificity currently found in the Universal Data Elements. More general values can be shared without any loss to reporting ability. Therefore, the Universal Data Elements should be revised to reduce the likelihood of recognition by the intimate stalker and/or data linkage threats by using the most general values possible. (For more information, see Section 11.)

Recommendation #35: The *date of birth* field should minimally be an *age range*. In fact, a Client may have more than one kind of age range specification. For example, there may be a data element related to 5-year age ranges, and another related to AHAR ranges (under 1, 1 through 5, 6 through 12, 13 through 17, 18 through 30, 31 through 50, 51 through 61, and 62 and over), enabling more reporting uses of the resulting data. (For more information, see Section 11.)

Recommendation #36: The *ZIP of last residence* field should be changed to either report the *first 3 digits of ZIP*, or even better, be changed to be a boolean flag denoting whether the Client's last residence was *within the geography covered* by the Planning Office or not. If the *first 3 digits of ZIP* are used, then only those values local to the Planning Office need be recorded. Clients from outside the local area would just have a special value, like 999, in order to prevent them appearing as unique outliers. (For more information, see Section 11.)

Recommendation #37: *PIN* should be removed. The Shelter should not provide its internal unique number. Instead, the Shelter should maintain an exact copy of the data provided so that records can be referred to in discussion with the Planning Office by the place (or row) in which the record appears. (For more information, see Section 11.)

Recommendation #38: Consider removing *Race* and *Ethnicity*. Experimental results showed that the addition of these fields increase risks to re-identification. (For more information, see Section 11.)

Recommendation #39: Shelters should consider renumbering *Household identification numbers* from 1 to the last household, prior to forwarding the information to the Planning Office. This makes sure the household identification number itself cannot be the basis for linking. (For more information, see Section 11.)

Recommendation #40: Replace the exact service dates (*Program Entry Date* and *Program Exit Date*) with number of days of service or with time periods (e.g., overnight, 2-14 days, 15-30 days, 30 plus days). (For more information, see Section 11.)

Recommendation #41: More sensitive data elements (such as *first name*, *Social Security number*, or full *date of birth*) may still be collected by Shelters in order to produce a useful UID. However, those values should continue to not be forwarded to the Planning Office as part of the Universal Data Elements. (For more information, see Section 11.)

1.8 Privacy assurance recommendations

In order to prevent collusion when the Planning Office and the HMIS consist of the same personnel, it is necessary to use additional privacy safeguards, beyond those for protecting UIDs (e.g. PrivaMix) and beyond merely changing the Universal Data Elements. Remedies involve expanding the post-processing done by PrivaMix so that the final dataset made available to the Planning Office contains either aggregate (not Client-level data) or provably anonymized Client-level data.

While PrivaMix guarantees privacy protection for UID creation and use in de-duplicating, linking vulnerabilities currently remain in the de-duplicated Universal Data Elements (Section 11). Problems stem from the selection of which data elements to associate with UIDs, and not from the UIDs themselves. Changes to the Universal Data Elements can help (Section 11), but such changes seem unable to be wholly satisfactory without effecting the usefulness of the de-duplicated data to the AHAR.

A PrivaMix System can anonymize de-duplicated results prior to forwarding data to the Planning Office. The anonymizaed data will not be vulnerable to linking, even if the Planning Office and HMIS collude.

At present, the PrivaMix Demonstration System, as used in the Iowa Experiment, de-duplicates Client information and then passes values associated with each UID to the Planning Office "as is." Instead of merely forwarding those values, a PrivaMix System could anonymize those data elements and then forward the anonymized results to the Planning Office.

There are numerous way to perform the anonymization. These include: replacing client-level results with pivot tables that show aggregate count information for combinations of data elements; replacing client-level data with an overall final report (e.g., the AHAR itself); or, provably anonymizing client-level data by automatically suppressing and generalizing values as needed. Each of these approaches can provide sufficient privacy protection, by replacing client-specific results with appropriately generalized ones. The result is privacy protection, even against data linking, and accurate de-duplicated results for the AHAR. (For more information, see Section 12.)

Recommendation #42: In order to prevent collusion when the Planning Office and the HMIS consist of the same personnel, it is necessary to use additional privacy safeguards, beyond those for protecting UIDs (e.g. PrivaMix) and beyond merely changing the Universal Data Elements. It is necessary to make sure the HMIS cannot link the Universal Data Elements to other service information contained in the HMIS. (For more information, see Section 11.)

Recommendation #43: Add post de-duplication anonymization to a PrivaMix System to make sure data provided to the Planning Office is not vulnerable to linking, even if the Planning Office and HMIS collude. The Planning Office receives provably anonymized de-duplicated results. (For more information, see Section 12.)

Recommendation #44: Consider having the final results be aggregate data only. Instead of Client-level data, a PrivaMix System can alternatively provide aggregate de-duplicated count distributions denoting how many Clients matched particular characteristics. An example of a count distribution are counts by age ranges. Distributions can involve more than one field to get more specific data. (For more information, see Section 8.2 and Section 12.)

Recommendation #45: Consider having the final results be the AHAR report itself. Instead of Client-level data, a PrivaMix System can alternatively provide the AHAR to the Planning Office. (For more information, see Section 8.2 and Section 12.)

Recommendation #46: Consider having the final results be anonymized Client-level data. Anonymized Client-level data generalizes or suppresses values, as needed, to protect privacy. Formal protection models identify which values to generalize or suppress from the resulting dataset so that each record ambiguously relates to a minimum number of people [30][31]. For example, if a 80 year old woman is an outlier in the data because of her age, either her age would be removed from the data or generalized to a category having more people, such as "50 plus" as appropriate value given the other ages appearing in the data. (For more information, see Section 8.2.6 and Section 12.)

In conclusion, PrivaMix provides an effective and accurate privacy-preserving means for constructing and de-duplicating UIDs. However, additional care with the Universal Data Elements must be taken to properly protect against unwanted data linkage with the HMIS. The problem is not with the UIDs but with the selection of data elements associated with the UIDs. A solution is to enhance a PrivaMix System to anonymize de-duplicated Client-level data and then forward the anonymized results to the Planning Office.

1.9 Summary of recommendations

Figure 2 below contains a quick summary of recommendations made. Some recommendations repeat because of the context in which it appears in the text.

#	Description	Section
1	Coordinate de-duplication across neighboring CoC's.	3.3
2	Not share Shelter PIN beyond Shelter.	3.5
3	De-duplicated results should not include PINs, UIDs, or Household IDs.	3.6
4	Shelters only include Clients who have left the Shelter.	4.1
5	Train personnel on accepted practices for handling Client data.	4.1
6	UIDs should be inconsistently assigned across Shelters.	4.2
7	Shelters should privacy notices for Client inspection.	4.2
8	Fields date of birth and ZIP should be less specific.	4.5, 7.1
9	Planning Office should delete any fields in the Universal Data Elements not needed.	4.6
10	Planning Office should sign Data Use Agreement with Shelters regarding linking.	4.6
11	Skilled person should certify System's risk of re-identification.	5.5
12	Skilled person should certify utility of de-duplicated results.	5.5
13	System using non-verifiable source information should instill trust.	6.9
14	System using encryption or hashing should use strong cryptographic methods.	6.9
15	System using encryption or hashing should control access to the function.	6.9
16	System using scan cards/RFID should avoid issuing multiple cards to the same Client.	6.9
17	UIDs should be removed from de-duplicated results.	6.9
18	Fields date of birth and ZIP must be less specific.	4.5, 7.1
19	System must satisfy VAWA's requirements limiting re-idenification.	7.2
20	A PrivaMix System must avoid Shelters producing the same UID for Clients.	8.1
21	Computers transmitting UDE over a network must adhere to accepted security standards.	8.1
22	If desirable, have a party other than the Planning Office orchestrate mixing.	8.2
23	A PrivaMix System should anonymize or aggregate results, rather than provide Client-level data.	8.2, 11
24	An economical PrivaMix System can result from using existing web browsers.	8.2
25	A PrivaMix Function must satisfy six noted requirements.	8.3
26	In a PrivaMix System. A Shelter value must be sufficiently large.	8.4
27	In a PrivaMix System, a Shelter should not even know its own private value.	8.4
28	In a PrivaMix System, unauthorized parties should be unable to use the Shelter's PrivaMix function.	8.4
29	In a PrivaMix System, Shelters should validate the number of UIDs requested to mix.	8.2, 8.4
30	In order to provide collusion with an HMIS, provide only aggregate or anonymized results.	8.4, 12
31	Shelters only include Clients who have left the Shelter.	8.4
32	UIDs should be removed from de-duplicated results.	8.4
33	Claims must be assessed for any particular PrivaMix implementation.	8.4
34	Make Universal Data Elements as general as remains useful to the AHAR.	11
35	Make date of birth field more general, such as the AHAR age classifications.	11
36	Make ZIP of last residence field more general, such as a boolean flag denoting whether in covered	11

#	Description	Section
	area.	
37	Remove PIN field from the Universal Data Elements.	11
38	Consider removing race and ethnicity fields from the Universal Data Elements.	11
39	Consider having Shelters renumber Household IDs to thwart any possible linking using the field.	11
40	Replace exact service dates with number of days or time periods.	11
41	Sensitive data elements may be used for UIDs, but not forwarded to the Planning Office.	11
42	Use privacy protections beyond UIDs and modified Universal Data Elements to thwart linking to HMIS.	11
43	Consider PrivaMix performing post de-duplication anonymization to thwart linking to HMIS.	12
44	Consider PrivaMix providing aggregate values, not Client-level data, to the Planning Office.	8.2, 12
45	Consider PrivaMix providing the AHAR itself, not Client-level data to the Planning Office.	8.2, 12
46	Consider PrivaMix providing anonymized Client-level data to the Planning Office.	8.2, 12

Figure 2. Summary of recommendations.

2. Need for an Unduplicated Accounting of Homeless Services

The number of homeless Americans appears to have dramatically increased in recent years, but no one actually knows the current number of homeless persons and counting them and understanding their service utilization patterns may not be as easy as it may first seem. At stake are resource allocations, program evaluations, and billions of dollars necessary for managing and resolving what may be one of the most serious social and economic crises of our time.

2.1 Examples of increases in the numbers of homeless Americans

Numerous anecdotal examples illustrate that the numbers of homeless Americans seem to be increasing over time and that related spending has reached dramatic heights.

HUD's Emergency Shelter Grants program funds resources for basic shelter and essential supportive services by awarding grants to state governments, large cities, urban counties, and U.S. territories. These awards totaled $10 million in 1987 and had grown to $115 million by 1997, with continued increases thereafter [3].

A report from the Northeast Ohio Coalition for the Homeless in 2005 that addressed the overflow of shelters in Cleveland Ohio, asserted that shelter costs in 2004 was 5.6 times the cost 10 years earlier for men and 9.4 times the cost 10 years earlier for women [4]. They predicted further increases over the next 10 years due to increased demand and warned that at the current rate of increased demand, county and city public sector funding will be exhausted.

A 2001 study of 27 U.S. cities reported that 37% of all requests for emergency shelters and 52% of all requests for emergency shelters from families were unmet in that year due to a lack of resources [5].

In April 2002, over 33,000 homeless people were provided emergency shelter each night by the New York City Department of Homeless Services [6]. This was the highest number they had recorded, and the cost of homelessness rose to record heights as well. According to a report by the New York City Independent Budget Office, New York City agencies spent almost $1 billion on homelessness in Fiscal Year 2001 [7].

Congress appropriated over $1 billion dollars to homeless assistance programs in the Fiscal Year 2002 HUD Appropriations Act [8].

2.2 Congress directs HUD to report on homeless service utilization

In response to noted increases in homelessness, which seem to reflect a growing social and economic crisis, Congress deemed it critical for the United States Department of Housing and Urban Development ("HUD") to work with local jurisdictions to develop an unduplicated accounting of homeless service utilization. Congress directed HUD to perform an unduplicated count[2] of homeless persons sufficient to provide annual reports to the Committee on

[2] The term "unduplicated count" is misleading. In ordinary language it tends to imply that the answer is a single number. In terms of the Congressional directive, it is actually an unduplicated accounting of shelter visits –i.e., the distinct visit patterns of each client across shelters.

Appropriations documenting the demographics and utilization patterns of homeless persons based on collected count data [8][9].

In the Fiscal Year 2002 HUD Appropriations Act, Congress allocated $2 million dollars specifically to continue work on a homeless data collection and analysis project that had begun the year before in the Fiscal Year 2001 HUD Appropriations Act [10]. This project seeks to document the demographics of homelessness, identify patterns in service utilization, and record the effectiveness of assistance programs. The work reported herein describes a way to achieve the unduplicated accounting within this data collection and analysis project.

2.3 Earlier attempts to count the number of homeless Americans

There have been previous attempts to count the number of homeless Americans by counting the number of people who are in shelters or on the streets at a given point in time.

On March 20, 1990, federal employees of the U.S. Bureau of the Census, in satisfaction of their duties as set forth in the U.S. Constitution, attempted to determine the exact number of Americans in the U.S. population by physically verifying the existence of each person, including an attempt to count every homeless person and gather related demographics [11]. Under this effort, termed Shelter-and-Street night, thousands of federal employees visited homeless shelters, inexpensive hotels, all-night eating establishments, bus stations, street corners and various urban places identified by local jurisdictions as places where homeless people are likely to be found. Employees were instructed not to ask who was homeless and not to awaken any persons found sleeping. Instead, they were told to count all visible persons (including children) found in these places and record demographics as either provided or as they appeared to the census taker. These efforts were able to add 240,140 homeless people to the official census count.

A more comprehensive estimate was provided by the Urban Institute using the 1996 National Survey of Homeless Assistance Providers and Clients [12]. The survey was designed to provide information about the providers of homeless assistance and the characteristics of homeless persons who used services by sampling 76 metropolitan and non-metropolitan areas, including small cities and rural areas at two points in the year. On a given night in February, 842,000 in 637,000 households were found homeless. On a given night in October, 444,000 people in 346,000 households were found homeless. Converting these point counts into a national annual projection, researchers at the Urban Institute estimated that between 2.3 and 3.5 million people were homeless in that year [13].

2.4 Limits of point-in-time counts

Point-in-time studies, like those mentioned above, give a limited static picture by only counting those who are homeless at specific places during a narrow slice of time. No explicitly-identifying person-specific information is necessarily collected, so double-counting can occur when clients use more than one service (i.e., appear at more than one point) during the capture period. An example is a client receiving meals at one facility and lodging at another during the same night; such a person may be counted once, twice, or not at all. Seasonal and climate variation may be missed altogether. Important differences in client circumstances may not be captured. For example, the frequency and lengths of time in which particular clients are in and out of homelessness is typically not captured by a point-in-time count. Prolonged unemployment, sudden loss of a job, lack of affordable housing, and domestic violence contribute to episodes of homelessness, while severe mental illness and addiction disorders often account for chronic homelessness. For these reasons, point-in-time studies may misrepresent the magnitude and nature of homelessness.

3. The HMIS Approach

In response to Congress' directive, HUD elected not to use the traditional point-in-time approach, but opted instead to develop and introduce national data and technical standards for locally situated computer systems that collect, process and share details of each client's utilization of service related to homelessness. These are termed Homeless Management Information Systems ("HMIS"), which are described in terms of the parties to and from which data flows and the data elements that constitute information flow. At this writing, the initial data elements had already been altered to better protect the privacy of domestic violence shelter clients from intimate abusers, but other privacy concerns remain which are addressed herein.

3.1 Data flow in HMIS

Using a HMIS, information does not flow directly from a homeless service provider to HUD. Instead, a HMIS introduces an intermediary (termed a "planning office" in this writing and referred to as a "continuum of care" or "CoC" in HUD documents)[3] that is local to a group of homeless service providers (e.g., shelters). The purpose of the planning office is to establish an HMIS for a group of service providers. Information flows from clients to service providers, who in turn, provide visit information to their local planning office. Because clients are expected to consume services from multiple providers, the planning office associates visits across providers over time to provide an unduplicated accounting to HUD for the services delivered in their geographical region.

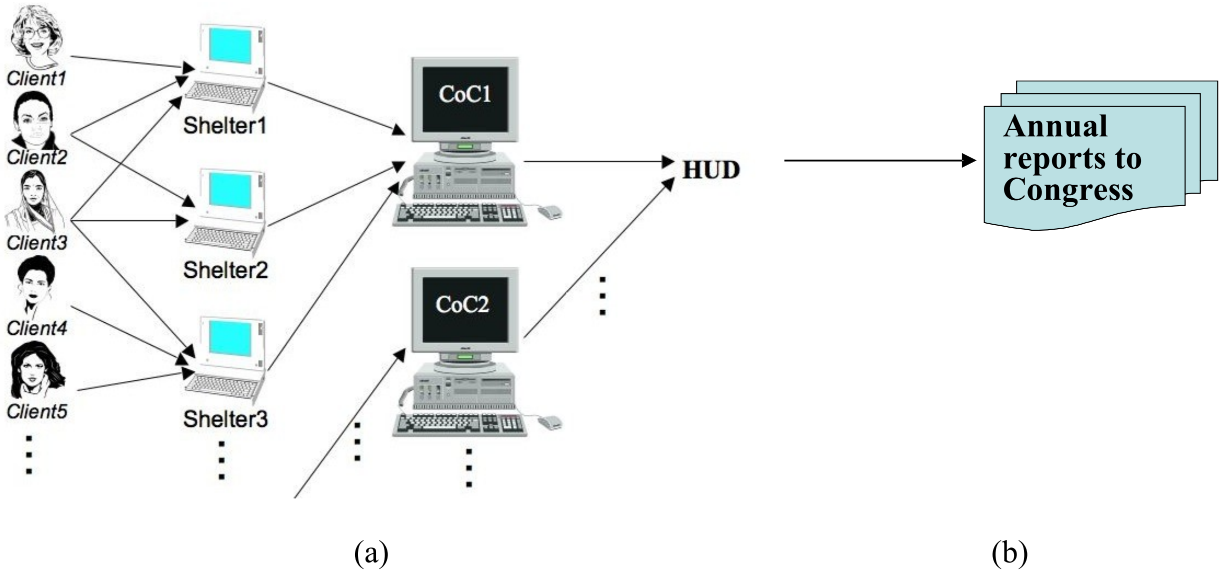


Figure 3. Flow of information from Clients to HUD: (a) Clients give information to Shelters, which report information to Planning Offices (CoCs), which in turn provide non-identifiable unduplicated count information to HUD, (b) which becomes the source data for annual homeless service utilization reports to Congress.

[3] The purpose of a regional planning office predates and is broader than HMIS, but for the purposes of this writing, planning offices are examined narrowly in their HMIS context.

While a HMIS includes lots of different services for many types of homeless clients, the work reported herein is specifically focused on clients who visit domestic violence shelters. Hereinafter, unless otherwise noted, references to "Shelters" are exclusively domestic violence shelters and may generally apply to a suite of homeless service providers. Similarly, references to "Clients" are homeless persons serviced by Shelters and to "Planning Offices" are the CoCs servicing Shelters.

Figure 3(a) depicts the flow of information from Clients to Shelters through Planning Offices to HUD. A Client visits one or more Shelters. Each Shelter provides information to one Planning Office. HUD uses non-identifiable information from Planning Offices to provide annual reports on the utilization patterns of homeless people to Congress; see Figure 3(b).

3.2 Comparing HMIS to point-in-time approaches

Because Client demographics and specific visit data are captured on each visit, many of the shortcomings found with point-in-time studies may potentially be resolved by the HMIS approach.[4]

For example, a HMIS seeks to record sufficient information to allow the same Client to be identified on subsequent visits to the same or other Shelters, thereby thwarting the potential for double counting. Associated date and length of stay information may be recorded to identify seasonal, climate and temporal visit patterns. Recording the reason given for each visit may help identify utilization characteristics related to different kinds of homelessness, and tracking Clients across the same and different shelters can provide recurrence and duration rates.

3.3 Concern about selecting planning offices

It is understood that a Client may visit one or more Shelters, which is why de-duplication across Shelters is necessary, but if the same Client visits Shelters reporting to different Planning Offices, then the de-duplication effort can be thwarted.

For example, consider Shelters servicing Boston and Cambridge Massachusetts. These are two cities between which people regularly walk and ride multiple times a day. If each of these cities has their own Planning Office, then a single Client being serviced by a Shelter in Cambridge and by another Shelter in Boston, would be counted twice –once by the Planning Office for Cambridge and again by the Planning Office for Boston. Similar situations can exist with Planning Offices located in close proximity to one another irregardless of city, county, or state boundaries. To combat this problem, the following recommendation is made.

[4] One shortcoming of both the survey used by the Urban Institute and the HMIS approach is the sole reliance on service providers. Homeless people who are not using shelters or covered services are not captured. These include homeless people who may live in automobiles, make-shift housing (such as cardboard boxes or tents), or doubled-up situations.

Recommendation #1: Coordination of privacy protection schemes is necessary across Planning Offices that service a geographical region in which Shelters within the region report to different Planning Offices but service some of the same Clients. Lack of coordination can distort the unduplicated accounting.

In 2006, HUD funded about 400 Planning Offices. This funding extends beyond HMIS to the coordination and funding of homeless services at the local level. A Planning Office defines its own geographical service area and competes to receive HUD funds for homeless programs. Because geographical service areas are not dictated by HUD, cooperative coordination of privacy protection schemes in overlapping areas allows a Client's utilization pattern to be determined without compromising the identity of the Client.

3.4 Removal of explicit identifiers from HMIS

Almost as soon as the first HMIS standards were announced, privacy concerns emerged over the need for protections for clients of domestic violence shelters [14][15]. Tracking victims of intimate domestic violence who seek refuge in Shelters may be necessary for HMIS accounting, but many feared HMIS data collection and sharing might become a vehicle to further endanger a victim whose information would appear in HMIS data as a result of her attempting to remove herself from a harmful situation.

A privacy protective action taken by HUD involved changing HMIS standards to allow Shelters to provide Client information without making reference to any client explicit identifiers (e.g., name and Social Security number). Instead, an approved proxy, coded, encrypted, hashed, or other alternative termed a "**unique identification number**" (or "**UID**") is to be used by Shelters to provide client information to Planning Offices, provided each Planning Office has the ability to recognize the occurrence of the same clients in the same and different shelters (including shelters that are not domestic violence provider shelters) over time.

Section 4 examines the nature of privacy threats in detail. Section 5 provides a method for assessing technologies for creating and using UIDs. Section 6 and Section 7 report on assessments of UID technologies initially considered. The remainder of this section examines the data elements collected and shared in a HMIS.

3.5 Details of HMIS data elements

HUD requires certain data elements be sent from Shelters to Planning Offices. The data elements that HUD requires Shelters to provide to Planning Offices are termed the "Universal Data Elements," and consists of a record for each Client's visit to a Shelter and includes the Client's UID. The original data elements were modified to use UIDs, in lieu of explicit identifiers, as shown in Figure 5. Shelters participating in a HMIS must collect the Universal Data Elements and share them with the Planning Office at least once a year in a privacy-preserving manner that includes replacing name and Social Security number with UIDs.

"Program-Specific Data Elements" are additional fields of information that Shelters may be required to provide on each Client visit. All McKinney Vento funded Shelters that are required to complete an Annual Progress Report are required to collect and share certain Program-

Specific Data Elements with the Planning Office[5]. Figure 6 lists the Program-Specific Data Elements and identifies which data elements are required for the Annual Progress Report.

HUD places no further restriction on the information collected between Clients and Shelters. Beyond the noted data elements, Shelters may elect to collect additional information for their own purposes. A Unique Person Identification Number ("PIN") is included among the Universal Data Elements. This field allows a Shelter to store its internal reference number for a Client. However, care must be taken to share only when the PIN is sufficiently privacy-protecting, as noted in the following recommendation.

Recommendation #2: *A Shelter may assign a unique person identification number (PIN) to internally identify a client, but it should not share the client's PIN externally. PINs that include the Client's name, Social Security number, or other characteristic may be used alone or in combination with other data elements to re-identify a Client. Any characteristic not allowed as a data element or a UID, should not be used as an externally shared PIN.*

In summary, Figure 4 shows the flow of information from a Client through the Planning Office to HUD using the Universal and the Program-Specific Data Elements.

Hereafter, the information transmitted from a Shelter to a Planning Office is collectively termed the "Dataset" in this writing and refers to the Universal Data Elements unless otherwise stated.

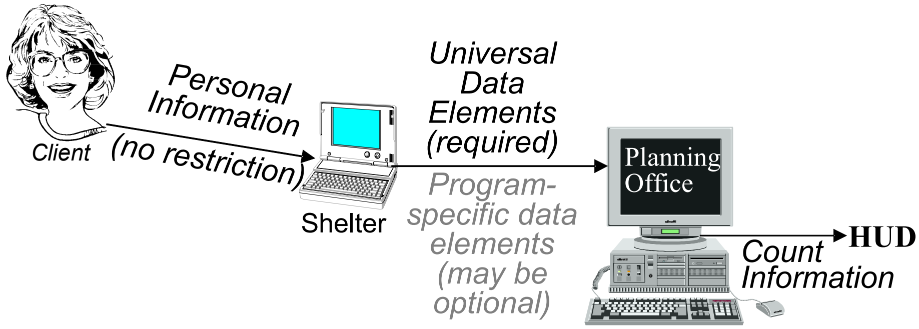


Figure 4. Flow of information: Client gives explicit personally identifying information to the Shelter, which provides the Universal Data Elements and Program-Specific Data Elements to the Planning Office, which in turn provides to HUD, non-identifiable, unduplicated count information of Client visits across all Shelters in the Planning Office's region.

[5] See http://www.hud.gov/offices/cpd/homeless/apr.doc.

UNIVERSAL DATA ELEMENTS		
#	Description	Comments and Possible Values
~~1~~	**~~Name~~**	**DV shelters collect but not share; use UID instead**
2	**~~Social Security Number~~**	**Domestic violence (DV) shelters collect but not share.**
3	Date of Birth	Month, day and year of birth
4	Ethnicity and Race	Hispanic/Latino or not; American Indian, Asian, Black, Pacific Islander, White
5	Gender	Male or female
6	Veteran Status	Yes, no, don't know, refused
7	Disabling Condition	Yes, no, don't know, refused
8	Residence Prior to Program Entry	Part I: Type of Residence Emergency shelter, transitional house for homeless, permanent housing for former homeless, psychiatric facility, substance abuse treatment facility, hospital (non-psychiatric), legal incarceration, rental unit, home ownership, family member's home, friend's home, emergency shelter voucher at hotel, foster care home, place not intended for habitation, other, don't know, refused
		Part II: Length of Stay in Previous Place Emergency shelter, transitional house for homeless, permanent housing for former homeless, psychiatric facility, substance abuse treatment facility, hospital (non-psychiatric), legal incarceration, rental unit, home ownership, family member's home, friend's home, emergency shelter voucher at hotel, foster care home, place not intended for habitation, other, don't know, refused
9	ZIP Code of Last Permanent Address	5-digit code, don't know, refused
10	Program Entry Date	Month, day, year
11	Program Exit Date	Month, day, year
12	**Unique Person Identification Number**	**"PIN" Shelter's internal reference number for Client.**
13	Program Identification Number ("Shelter ID")	Part I: FIPS code identifying geographic location of shelter
		Part II: Identification code for shelter, including HUD assignment
		Part III: Program Type Code: Emergency shelter, transitional housing, permanent supportive housing, street outreach, homeless prevention service, other service
14	Household Identification Number	Constructed number to identify clients receiving services as a household

Figure 5. HMIS Universal Data Elements includes the generated unique identification number (UID).

PROGRAM-SPECIFIC DATA ELEMENTS

#	Description	Need for Annual Progress Report	Comments and Possible Values
1	Income and Sources	Yes	Part I: Source of Income Earned income, unemployment insurance, supplemental security income (SSI), Social Security disability (SSDI), veteran's disability, private disability insurance, worker's compensation, temporary assistance for needy families (TANF), general assistance program (GA), Social Security retirement income, veteran's pension, former job pension, child support, alimony, other source, no financial resources.
			Part II: Total monthly income in dollars
2	Non-cash benefits	Yes	Food stamps, MEDICAID health insurance, MEDICARE health insurance, state children's health insurance, women-infants-children program (WIC), veteran's medical services (VA), TANF child care, TANF transportation services, other TANF services, public housing, other source.
3	Physical Disability	Yes	No, yes
4	Developmental Disability	Yes	No, yes
5	HIV/AIDS	Yes	No, yes
6	Mental Health	Yes	Part I: Mental health problem – no, yes
			Part II: Expected indefinite duration – no, yes
7	Substance Abuse	Yes	Part I: Problem: none, alcohol, drug, dully diagnosed
			Part II: Expected indefinite duration – no, yes
8	Domestic Violence	Yes	Part I: Experience –no, yes
			Part II: Time of experience past 3 months, 3-6 months ago, 6 to 12 months ago, more than a year ago, don't know, refused.
9	Services Received	Yes	Part I: Date of service – month, day, year
			Part II: Type of Service Food, housing, material goods, financial aid, transportation, consumer assistance, legal services, education, health care, HIV/AIDS services, mental health care, substance abuse services, employment, case management, day care, personal enrichment, outreach, other.
10	Destination	Yes	Part I: Destination Emergency shelter, transitional housing, permanent housing for formerly homeless, psychiatric facility, substance abuse treatment center, hospital (non-psychiatric), legal incarceration, rental unit, home own, family home, friend's home, hotel paid by shelter voucher, foster care, place not meant for habitation, other, don't know.
			Part II: Tenure Refused, permanent, transitional, don't know, refused
			Part III: Subsidy Type None, public housing, Section 8, S+C, HOME program, HOPWA program, other housing subsidy, don't know, refused.
11	Reasons for Leaving	Yes	Housing opportunity, completed program, non-payment of rent, non-compliance with project, criminal activity, reached maximum

PROGRAM-SPECIFIC DATA ELEMENTS			
#	Description	Need for Annual Progress Report	Comments and Possible Values
			time allowed, needs could not be met, disagreement with rules or people, death, disappeared, other
12	Employment	No	Part I: Employed – no, yes
			Part II: If employed, number of hours worked past week
			Part III: If employed, tenure --permanent, temporary, seasonal
			Part IV: If not employed ,looking for work – no, yes
13	Education	No	Part I: In school – no, yes
			Part II: Received vocational training – no, yes
			Part III: Highest Level of School Completed No schooling, nursery school to 4th grade, 5th or 6th grade, 7th or 8th grade, 9th grade, 10th grade, 11th grade, 12th grade with no diploma, high school diploma, GED, post-secondary school.
			Part IV: Post-Secondary Education If high school diploma or equivalent, earned Associated Degree, Bachelor's, Masters, Doctorate, other graduate/professional degree.
14	General Health Status	No	Excellent, very good, good, fair, poor, don't know
15	Pregnancy Status	No	no, yes
16	Veterans Information	No	Part I: Military Service Era Persian Gulf, post Vietnam, Vietnam era, between Korean and Vietnam wars, Korean war, between WWII and Korean war, World War II, between WWI and WWII, World War I.
			Part II: Duration of active duty in months
			Part III: Served in a war zone – no, yes
			Part IV: If served in War Zone, Specify Zone Europe, North Africa, Vietnam, Laos and Cambodia, South China Sea, China-Burma-India, Korea, South Pacific, Persian Gulf, other.
			Part V: If served in war zone, number of months served
			Part VI: Received hostile or friendly fire –no, yes
			Part VII: Branch of the Military Army, Air Force, Navy, Marines, other.
			Part VIII: Discharge Status Honorable, general, medical, bad conduct, dishonorable, other.
17	Children's Education	No	Part I: Current enrollment status – no, yes
			Part II: Name of School (explicitly stated)
			Part III: Type of School – public, parochial-private
			Part IV: Last date of enrollment –month, day, year
			Part V: If not enrolled, Identify Problem Residency requirements, availability of school records, birth certificate, legal guardian requirements, transportation, lack of preschool program, immunization requirements, physical examination requirements, other.

Figure 6. Program Specific Data Elements are supplemental information that may be made available to planning offices.

3.6 The unduplicated accounting

The motivation for HMIS data collection and sharing are the annual reports HUD will provide to Congress, which will report on homeless demographics, utilization patterns, and service availability. These reports are termed the "**Annual Homeless Assessment Report**" ("**AHAR**"). To produce the AHAR, Planning Offices use HMIS data to provide aggregate count information to HUD.

HUD provided the first AHAR to Congress in 2006 using HMIS data collected in 2005. An initial draft of the data analysis for the 2006 AHAR shows how HMIS data elements contribute to the AHAR [19]. Basic questions addressed by the AHAR focus on emergency shelters and transitional housing for individuals and for households. Figure 7 has a sample of the kinds of questions answered by the AHAR using HMIS data elements. The sample questions pertain to individuals at emergency shelters, but similar questions exist for transitional housing and for households. Notice that all the data elements are used except UID and PIN (recall name and Social Security number had already been removed). A Planning Office provides HUD with answers to these questions, which are aggregated counts and not the raw data used to compute the counts.

A Planning Office can generate a "De-identified Dataset"[6] to perform the de-duplication and compute the unduplicated count information needed for the AHAR by linking Client demographics to Shelter utilizations using Client UIDs. The resulting data, which does not itself have to further include Client UIDs and PINs, is de-identified.

The UID is used to identify data relating to the same Client. Once the visit records are grouped by Client, the UIDs are no longer needed. A sequentially assigned Client number from 1 to the total number of distinct Clients appearing in the dataset can be used to reference Clients in the De-identified Dataset.

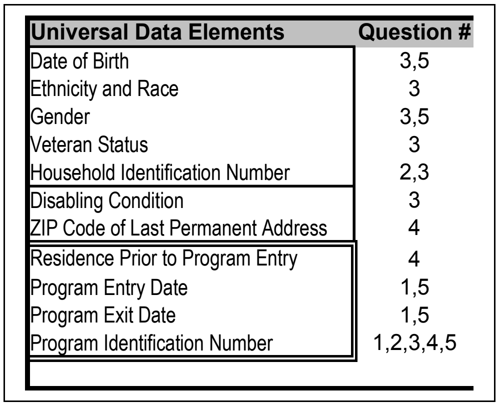
PINS are not needed in the De-identified Dataset. If a data problem occurs, the Planning Office has the originally received data for communicating with a Shelter using the Shelter's PIN.

Similar to UIDs, once Clients belonging to the same households are linked together, the Household Identification Number can be replaced with a sequentially assigned number from 1 to the total number of distinct households appearing in the dataset.

Figure 8 shows an example for a single Client. The Client's utilizations relate to her demographics but not to her explicit identity. Clients belonging to the same household are linked by sharing the same Household Identification number. Figure 9 provides an example of four clients, two of which are in the same household. The de-identified data can be used to compute values necessary to forward to HUD for the AHAR. Removing PINs and replacing UIDs and Household Identification numbers adds privacy protection to the De-identified Dataset, though more privacy protections are needed, as discussed in the remainder of this writing.

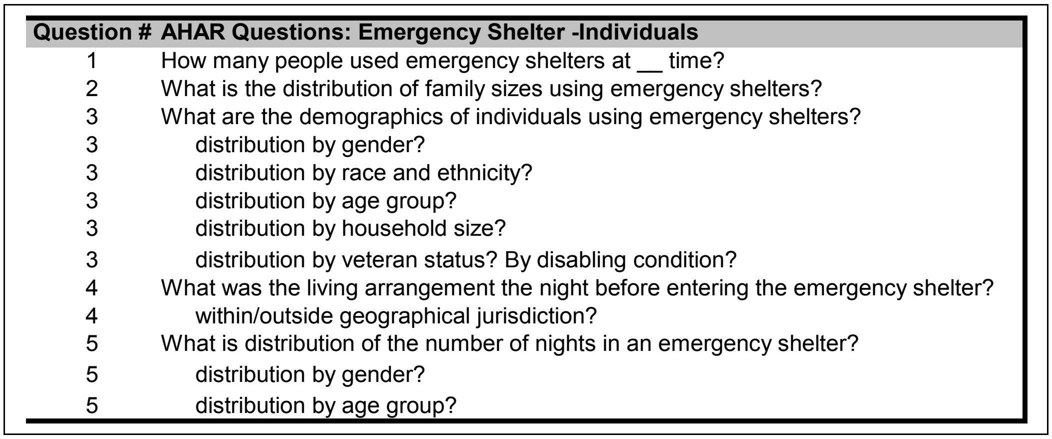
[6] While the De-identified Dataset is sufficient for computing the aggregate unduplicated count information that is forwarded to HUD, Planning Offices are not required to use the exact de-identified dataset described above.

Recommendation #3: *If a Planning Office produces a De-identified Dataset from the HMIS data collected from Shelters, the De-identified Dataset should not include any original Personal Identification Numbers (PINs), Unique Identification numbers (UIDs), or Household Identification numbers.*



Universal Data Elements	Question #
Date of Birth	3,5
Ethnicity and Race	3
Gender	3,5
Veteran Status	3
Household Identification Number	2,3
Disabling Condition	3
ZIP Code of Last Permanent Address	4
Residence Prior to Program Entry	4
Program Entry Date	1,5
Program Exit Date	1,5
Program Identification Number	1,2,3,4,5

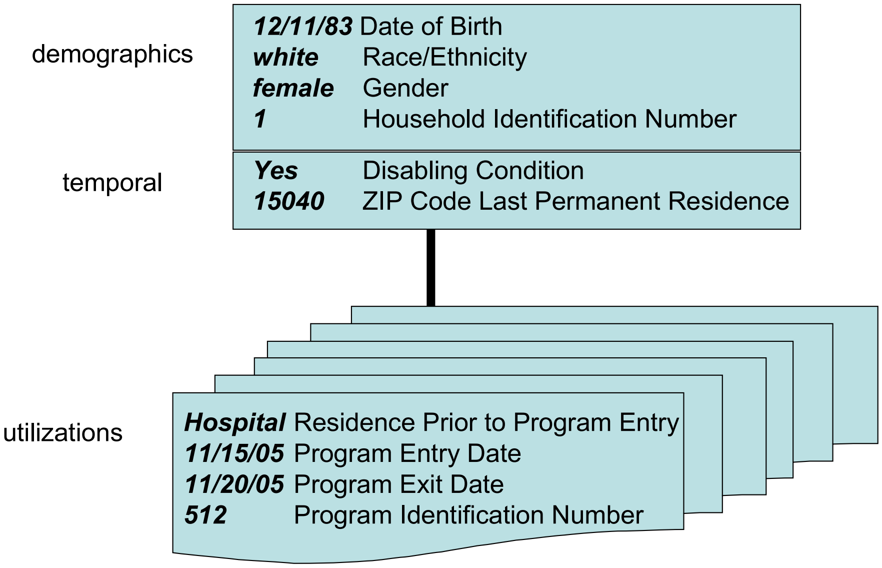
(a)



Question #	AHAR Questions: Emergency Shelter -Individuals
1	How many people used emergency shelters at __ time?
2	What is the distribution of family sizes using emergency shelters?
3	What are the demographics of individuals using emergency shelters?
3	distribution by gender?
3	distribution by race and ethnicity?
3	distribution by age group?
3	distribution by household size?
3	distribution by veteran status? By disabling condition?
4	What was the living arrangement the night before entering the emergency shelter?
4	within/outside geographical jurisdiction?
5	What is distribution of the number of nights in an emergency shelter?
5	distribution by gender?
5	distribution by age group?

(b)

Figure 7. Data elements from Figure 5 above (a) associated with sample questions answered by the AHAR (b). Planning Offices provide HUD with aggregated unduplicated count information as answers to the questions.

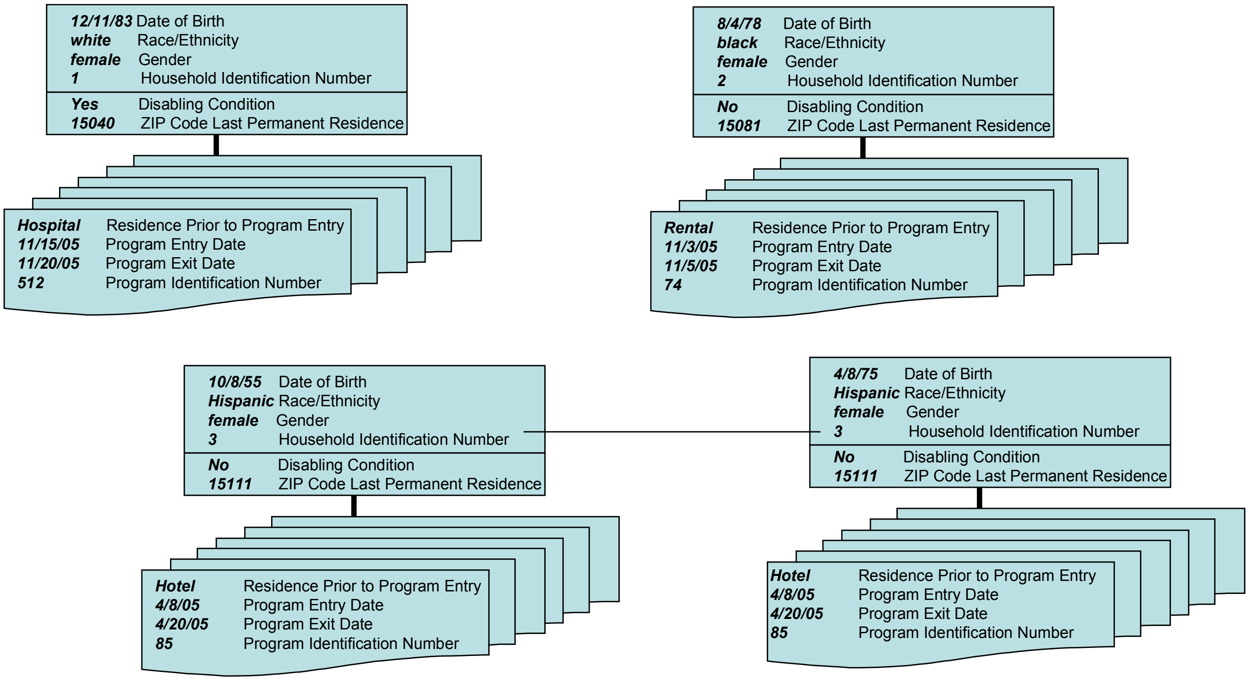


12/11/83 Date of Birth
white Race/Ethnicity
female Gender
1 Household Identification Number

Yes Disabling Condition
15040 ZIP Code Last Permanent Residence

Hospital Residence Prior to Program Entry
11/15/05 Program Entry Date
11/20/05 Program Exit Date
512 Program Identification Number

Figure 8. De-identified data for a Client includes demographics, some information that may change over time (disabling condition and ZIP of last residence), and program utilizations.



12/11/83 Date of Birth
white Race/Ethnicity
female Gender
1 Household Identification Number
Yes Disabling Condition
15040 ZIP Code Last Permanent Residence

Hospital Residence Prior to Program Entry
11/15/05 Program Entry Date
11/20/05 Program Exit Date
512 Program Identification Number

8/4/78 Date of Birth
black Race/Ethnicity
female Gender
2 Household Identification Number
No Disabling Condition
15081 ZIP Code Last Permanent Residence

Rental Residence Prior to Program Entry
11/3/05 Program Entry Date
11/5/05 Program Exit Date
74 Program Identification Number

10/8/55 Date of Birth
Hispanic Race/Ethnicity
female Gender
3 Household Identification Number
No Disabling Condition
15111 ZIP Code Last Permanent Residence

Hotel Residence Prior to Program Entry
4/8/05 Program Entry Date
4/20/05 Program Exit Date
85 Program Identification Number

4/8/75 Date of Birth
Hispanic Race/Ethnicity
female Gender
3 Household Identification Number
No Disabling Condition
15111 ZIP Code Last Permanent Residence

Hotel Residence Prior to Program Entry
4/8/05 Program Entry Date
4/20/05 Program Exit Date
85 Program Identification Number

Figure 9. De-identified data for Clients includes utilization patterns. Some Clients are linked together by sharing the same Household Identification Number (depicted by the link between the bottom Clients).

4. Privacy Threats

There are two primary motivations for infringing on the privacy of Clients of domestic violence shelters: the intimate abuser seeks to learn the physical location of the Client; and, the Planning Office seeks to link Client information to other available data to learn more about Clients overall. The next sections discuss these threat models in detail.

4.1 Intimate stalker threat

Domestic violence shelters have historically had to protect Clients from intimate and aggressive abusers and concerns are well founded. Over 31% of all women[7] murdered in the United States are murdered by husbands, boyfriends , or exes – the majority killed after attempting to leave an abusive relationship [16][17]. The National Institute of Justice estimated that 73% of domestic violent assaults go unreported largely because of women's lack of faith in the system [17].

Personal stories are quite chilling. As an example, consider a case from Los Angeles, California [18]. In 2001, a woman's husband was unemployed and had been drinking heavily. When she refused to have sex with him, he attacked her, prevented her from calling for help, and held her captive in her home. Various other incidents recurred. Eventually she was able to get a spot in a family shelter for herself and her two children. After leaving the shelter, the husband quickly tracked her down and strangled her to death with a belt.

The "intimate stalker" (an name given in this writing to an intimate abuser who stalks a Client) challenges computer systems that record and share Client visit information in several ways. First, the intimate stalker typically has knowledge of various personal facts about the Client that may be recorded in data held by the Shelter in which the victim resides. For example, an intimate stalker is likely to know the victim's name, date of birth and Social Security number, which may not be readily known by the general population. Second, the intimate stalker tends to be highly motivated to locate a targeted Client. For example, repeated violations of court orders and police reports describing escalating incidents of death threats, stalking and harassment are common. Finally, an intimate stalker may use insider access (either his own or by compromising an insider who has access to the data) to gain location information on a targeted Client. For example, an intimate stalker may persuade a family member or a friend to assist in revealing a Client's Shelter location by expressing a desire to reconcile for the sake of children or because situations (such as obtaining a new job) have changed.

No one solution addresses all these concerns, however some recommendations can be made immediately and others will be made in subsequent sections.

[7] While the wording used has a bias that women are victims and men are abusers, it is important to note that men are also victims and that abusers can be male or female.

One recommendation, as stated below, is to thwart the intimate stalker's ability to locate the Client by making sure visit information shared with the Planning Office is no longer current. This protection is not a first line of defense against an intimate stalker and should not be the only protective action taken. It merely offers supplementary protection. Stronger protections, which will be examined later in this writing, guarantee that the location of any Shelter in which the Client has historically visited cannot be learned by the intimate stalker. Stronger protection is important because some Clients tend to re-visit the same Shelters and an intimate stalker's knowledge of a historic visit can pose future problems.

Recommendation #4: *A Shelter should release Client information to the Planning Office some time after the Client has left the shelter.*

Another recommendation, as stated below, is aimed at helping thwart the stalker's ability to recruit or compromise those with insider access to Client information. This protection only provides supplemental protection. Stronger protections, which are examined later in this writing, guarantee that the Client's information cannot be found in information shared or stored external to the Shelter.

Recommendation #5: *Shelters and planning offices should train personnel on the responsibilities and accepted practices for collecting, storing and sharing client information.*

4.2 Data linkage threat

Beyond the intimate stalker threat in which information about a single Client is sought, the data linkage threat involves learning information about most, if not all, Clients by matching the information to other available data in order to use HMIS data inappropriately. This kind of activity is most likely to occur at Planning Offices where linking can be used to learn information about a larger number of Clients than those at just one Shelter. Protecting privacy in this setting cannot involve thwarting all linking, because the HMIS de-duplication task the Planning Office performs on the data requires linking records that belong to the same Client across Shelters. Instead of thwarting all linking, privacy protection in the HMIS setting involves thwarting linking attempts that may re-identify Clients.

Figure 10 provides an example in which the Dataset is linked to publicly available voter information on {*ZIP*, *date of birth*, *sex*} to re-identify the records in the Dataset by *name*. The more uniquely occurring {*ZIP*, *date of birth*, *gender*}, the more fruitful the re-identifications.

Most UIDs are designed to be uniquely assigned to Clients, so as a result, UIDs can also be used as the basis for linking datasets. That is not surprising given that HUD introduced UIDs into HMIS in order to link Client visits. However, if the same UIDs are also used with non-HMIS data, then they become the basis for linking HMIS data beyond the HMIS context. The following recommendation is aimed at thwarting secondary uses of HMIS data using UIDs.

Recommendation #6: *UID values assigned to Clients of domestic violence shelters should not be used (i.e., stored or referenced) by any non-HMIS program to which the Clients may participate to limit unwanted linking.*

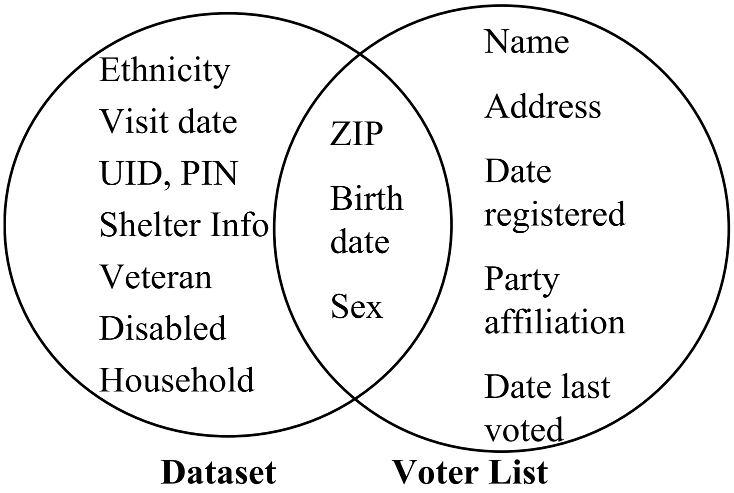


Figure 10. Example of linking Dataset to a publicly available population register, such as voter list, to re-identify the names of Clients appearing in Dataset.

In some cases, Planning Offices may decide to use HMIS data outside the HMIS context and in so doing, may purposefully link HMIS data to other non-HMIS data, even though this is unnecessary to achieve HMIS objectives. UID technologies can be constructed to thwart this behavior, as discussed later in this writing, but if this activity is desired, then Clients and Shelters should be made aware of this practice and any increased risk that may result. This is the motivation behind the following recommendation.

Recommendation #7: Shelters and Planning Offices are already required to issue and post privacy notices to clients about the data collection, sharing, and linking practices of the shelters and planning offices in which the client's data will be part [1]. Beyond the role this requirement plays as a Fair Information Practice, this requirement is also important to help ensure the integrity of the information a client provides in forming the client's UID.

4.3 Re-identification

A "re-identification" results when a record in Dataset can reasonably be related to the Client who is the subject of the record in such a way that direct and rather specific communication with the Client is possible. Figure 11 provides a depiction of a re-identification in which external information is linked on month and year of birth (9/1960), gender (F), and ZIP code (37213) to identify the visit information as belonging to Ann. The re-identification is sufficient to send a letter to Ann's residence.

For another example, consider Figure 10 in which Dataset is linked to a voter list to re-identify Client visits by name, even though Client names had been omitted from the visits in an attempt to protect privacy (recall Section 3.4.3).

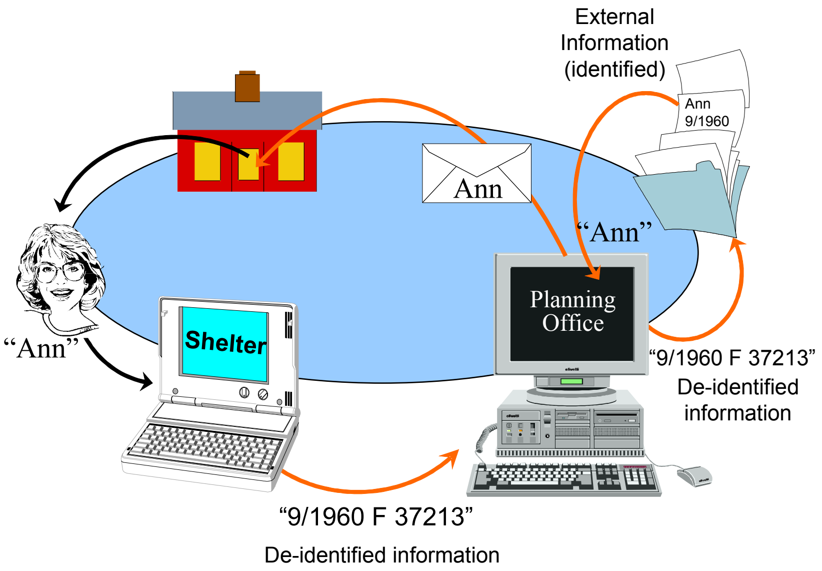

External Information (identified)
Ann
9/1960
Ann
"Ann"
Planning Office
"9/1960 F 37213"
De-identified information
"Ann"
Shelter
"9/1960 F 37213"
De-identified information

Figure 11. Depiction of re-identification. Ann leaves her home and gives her explicitly identified information to the Shelter. De-identified information about Ann is provided to the Planning Office, but in this depiction, the information can be used with external information (or personal knowledge) to re-identify the information as belonging to Ann. A re-identification occurs if there is sufficient information to directly communicate with Ann (not limited to mail), shown in the diagram as mailing an envelope to her original residence (or alternatively, sending the letter to Ann at the Shelter in which she resides).

4.4 Identifiability

One way to report the risk of re-identification is to determine the number of people to whom a record could refer. This is termed "identifiability." Figure 12 shows two examples in which information is released and compared against a known population. On the left, Figure 12 (a), each of the released profiles are ambiguous in terms of head shape and shading. Neither can be uniquely identified. The top released profile matches Hal and Len indistinguishably and the bottom profile ambiguously matches Jim and Mel. The release shown on the upper right of Figure 12 (b) is different. There is only one person in the known population (Hal) having the same color and head shape. In this case, the record referring to Hal is uniquely re-identified even though many of Hal's details had been removed.

While unique re-identifications obviously pose a privacy problem, so do situations in which a record maps ambiguously to a few known people. In Figure 12(a), both released profiles map to two individuals, but these people are both explicitly known, so they can both be contacted with little effort. Of course, the larger the number of people to whom a record refers, even if all of the people are known, the greater the effort usually needed to contact so many or make use of the information.

Counting the number of possible re-identifications for a record is a useful measure of privacy risk, but what is needed is a way to estimate the number of people to whom a record might refer.

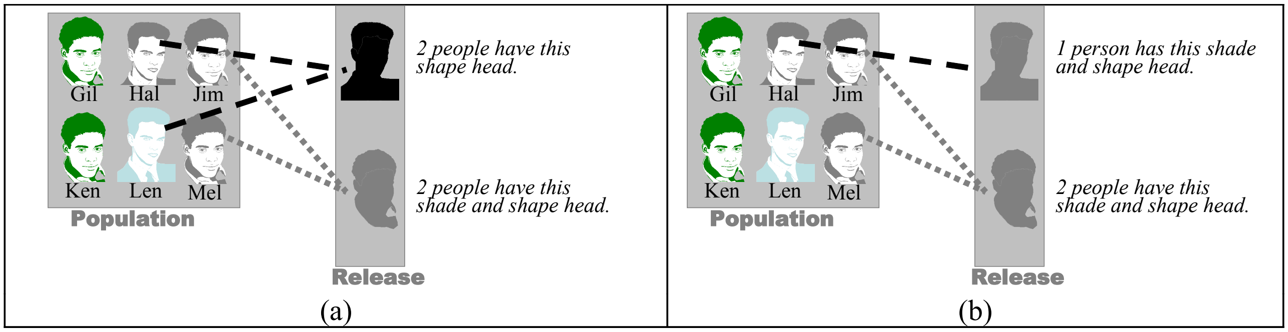

Gil
Hal
Jim
Ken
Len
Mel
Population
2 people have this shape head.
2 people have this shade and shape head.
Release
(a)
Gil
Hal
Jim
Ken
Len
Mel
Population
1 person has this shade and shape head.
2 people have this shade and shape head.
Release
(b)

Figure 12. The identifiability of the profiles released in (a) are each ambiguously re-identified to two named persons. The top profile released in (b) is uniquely re-identified to Hal.

4.5 Identifiability of a dataset

The Risk Assessment Server is a commercially available system that reports re-identification risks by estimating the number of named persons to which each record could relate given its model of the U.S. population and its knowledge of publicly available datasets [20]. The output of the Risk Assessment Server is a plot of identifiability estimates, in graduated size groupings, that report the number of people to which a released record is apt to refer.

Figure 13 shows the results from the Risk Assessment Server based on {*date of birth*, *gender*, *5-digit ZIP*} from Dataset. The lower left plot shows that 87% of the population are uniquely identified by these characteristics. As age information is generalized and as geographical reference to the Client's prior residence is made less specific, uniqueness deteriorates and privacy protection increases. For example, {*year of birth*, *gender*, *5-digit ZIP*} drops the unique identifiability to 0.04% (see the lower right plot in Figure 13).

Dataset currently requires Shelters to provide the full month, day and year of birth and all 5 digits of the Client's last residential ZIP code, yet the AHAR uses only gross age values and geography relative to Shelter's service area (refer to Section 3.6). The following recommendation is aimed at increasing privacy protection by changing the level of specificity in these fields.

<u>*Recommendation #8:*</u> *The fields* date of birth *and* ZIP code of last residence*, which are among the data elements HUD recommends HMIS collect in the Universal Data Elements, should contain information less specific than the month, day, and year of birth and all 5 digits of the ZIP (or postal) code.*

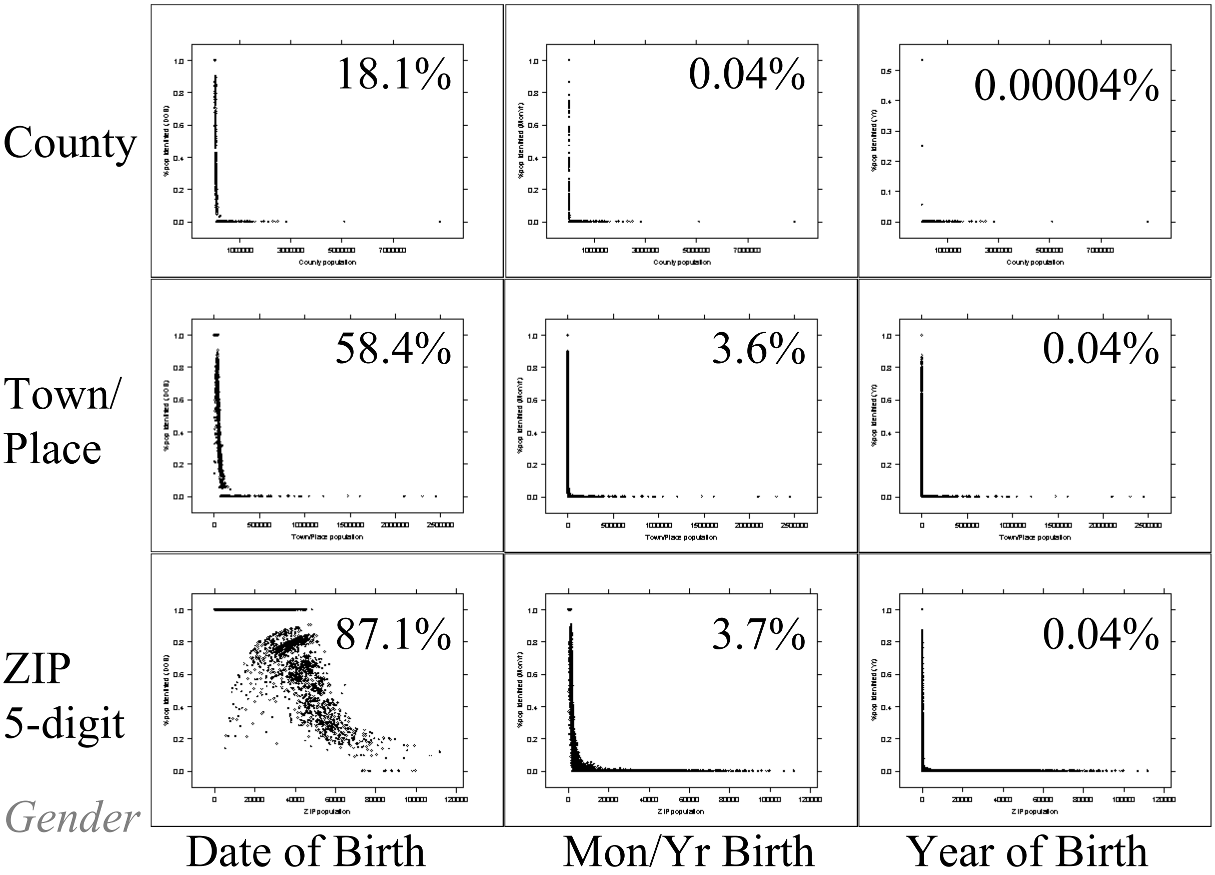


Figure 13. {*date of birth*, *gender*, *5-digit ZIP*} uniquely identifies 87.1% of USA population, but as ZIP is made less specific, the identifiability drops to 18.1% (bottom to top). Similarly, as the age of the client is made less specific, the identifiability drops to 0.04% (left to right). All values include gender. The horizontal axis of each sub-plot is the number of people who reside in the geographical area and the vertical axis is the percentage of the population uniquely identified by the noted combination of demographics noted. As the demographics are aggregated, the points move towards 0% identifiable.

4.6 Privacy concerns in Program-Specific Data Elements

Planning Offices that receive Program-Specific Data Elements (Figure 6) have some additional privacy concerns to consider to best protect Client data.[8] Program-Specific Data Elements may be linked to other available programmatic information to re-identify Clients. This vulnerability differs among municipalities and states as different kinds of secondary data from related programs are available.

A Planning Office is assumed to have multiple versions of data available, each having different re-identification risks and therefore different access policies. Figure 14 provides an overview. In terms of re-identification risk, the most sensitive data is that which first arrives at the Planning Office from the Shelter. These data may be separated into the Dataset used for the unduplicated accounting (the Universal Data) and the Program-Specific Data. No UIDs should appear in the Program-Specific Data. The De-identified Dataset is of least risk. A Planning Office may make internal access policies commensurate with these levels of risk. This advice regarding the maintenance of various versions of data is for consideration by Planning Offices and is not required.

Different versions of the data have different purposes. The originally received data could be maintained intact for quality control of Client information with Shelters (using PINs). The De-identified Dataset (modified to have less specific values of ZIP and date of birth) offers the least risk of re-identification and can be used to compute the unduplicated count information. In cases where the Shelter does not provide Program-Specific Data, the Dataset and the Originally Received Data are the same.

Recommendation #9: *A Planning Office may generate a "De-identified Dataset" from collected Shelter data to compute the unduplicated accounting. If so, the Planning Office should only use the Universal Data Elements in computing the De-Identified Dataset and remove (or obscure) elements from the De-identified Dataset that may appear in other data held by the Planning Office to limit secondary linking to other data held by the Planning Office.*

Recommendation #10: *Personnel in the Planning Office should sign a data use agreement with Shelters or provide notice to Shelters that either disallows the linking of the De-Identified Dataset to any other data or makes explicit the linking intended.*

[8] The requirements of the Program-Specific Data elements reside outside the scope of this work. However, some relative re-identification risk is noted.

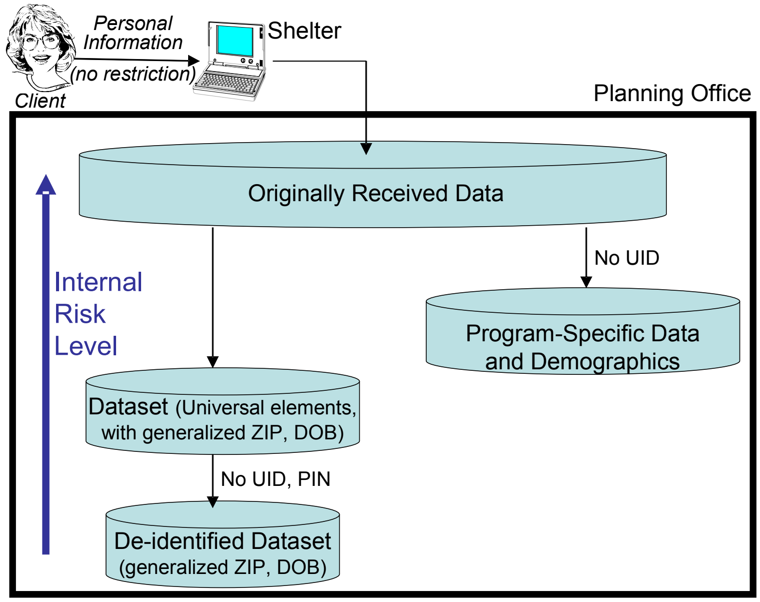

Personal Information
(no restriction)
Client
Shelter
Planning Office
Originally Received Data
No UID
Internal Risk Level
Program-Specific Data and Demographics
Dataset (Universal elements, with generalized ZIP, DOB)
No UID, PIN
De-identified Dataset (generalized ZIP, DOB)

Figure 14. Versions of data maintained by a Planning Office with relative internal risk of re-identification. The originally received data has the most internal risk and the De-identified Dataset has the least.

5. Assessing UID Technologies

Immediately after HUD introduced a UID in the Universal Data Elements ("Dataset"), many Planning Offices and Shelters began exploring technologies to construct, maintain, and use UIDs. The goal of this section is to describe how to assess plans and technologies in terms of their ability to perform an unduplicated accounting while protecting privacy. This section itemizes what should be the content of the assessment and what problems it should address.

This section and the next examine initial UID technologies in the absence of more recent regulation ("VAWA"). VAWA, as discussed in Section 7. subsequently rendered most of these technologies unacceptable. Section 8 through Section 12, introduces and tests PrivaMix as a solution that meets the higher privacy standards imposed by VAWA, but these two sections remain useful in characterizing the space of what makes a solution acceptable..

In this writing, a "Proposed Solution" is a UID technology bundled with an accompanying set of policies and practices for the construction, maintenance and use of a UID technology for Clients of Shelters in a HMIS. The entire package, UID technology, policies and practices, bundled together, is the subject of the assessment.

The overall problem for which UIDs have been introduced is easy to understand. It is termed the "HMIS Unduplicated Count Problem" and is stated below.

The HMIS Unduplicated Count Problem.
Given a set of Clients, a set of Shelters, and a Planning Office, where Clients visit Shelters, and Shelters report Dataset to the Planning Office on the Clients that visit, how should information about Clients be reported to Shelters and to the Planning Office such that the Planning Office can identify distinct visits of Clients across Shelters but not the identities of the Clients?

In order to determine whether a Proposed Solution is a sufficient solution to the HMIS Unduplicated Count Problem, an assessment must be done that demonstrates that the Proposed Solution remains useful for HMIS purposes while still being minimally invasive to privacy. Framed this way, the HMIS Unduplicated Count Problem is an optimization problem.[9] On the one hand, a Proposed Solution should provide an accurate accounting of distinct Client visits. On the other hand, a Proposed Solution should protect the privacy of Clients. The sufficiency of a Proposed Solution is based on performance guarantees that can be made. Specifically, a performance guarantee that the Proposed Solution has a minimal risk of re-identification when the solution is considered with other publicly and readily available information and techniques is termed a "Compliance Statement" in this writing. Similarly, a performance guarantee that the Proposed Solution provides a reasonably accurate unduplicated accounting of client visit patterns to shelters within the regional setting it is to be deployed is a "Warranty" in this writing. An assessment of a Proposed Solution is done by providing Compliance and Warranty statements.

The next subsections provide more information about Warranty and Compliance statements. But first, the notion of "source information" and "de-duplication instrument" are introduced.

[9] Viewing the HMIS Unduplicated Count Problem as an optimization of privacy and utility is deemed unacceptable by VAWA, which happened after efforts to construct UID technologies had begun.

5.1 Basic terms

A UID technology involves transforming some source information collected from the Client at a Shelter, into a UID. The ideal is to have a UID uniquely associated with a Client such that no two UIDs relate to the same Client, and a Client has only one UID. Resulting UIDs are used by the Planning Office to identify the same Clients across Shelter visits by matching UIDs or by using a "de-duplication" instrument. These terms are further described in the next subsections.

5.1.1. Source information

Source information is something a Client holds or knows that forms the basis of the Client's UID. Common examples of source information are name, date of birth, and Social Security number. The source information is not the same as the UID, but instead is used as the basis for a method (or algorithm) that computes a UID from it. For example, an algorithm for constructing a UID could involve concatenating the Client's date of birth with the first 4 letters of the Client's first name. For example, Alice with birthdate 9/12/1960 would have UID "09121960ALIC."

In some cases, the source information may rely solely on volunteered verbal information from the Client. This is termed "non-verifiable" source information. Client information is just accepted as stated and is not checked against other credentials.

An interesting example of non-verifiable source information for UIDs is realized by allowing a Client to makeup her own UID (e.g., "100678") or by constructing a UID based on Client answers to simple questions like "your favorite color, song, and ice cream" or "which picture most resembles your first love." As long as the Client answers consistently across multiple Shelter visits, the UID will be associated with the Client. As long as the questions tend to evoke unique answers from each Client and Clients answer the same way on each visit, then the resulting UID will be uniquely associated with a Client.

"Verifiable" source information is something provided by the Client that can be confirmed. Examples include a driver's license or a fingerprint.

5.1.2. De-duplication instrument

A set of algorithms that describe how to construct a UID from source information and how to use UIDs to match Clients are collectively termed a "UID technology." Algorithms that construct UIDs may be as simple as concatenating parts of Client demographics, as demonstrated above, or more complicated as computing a unique value for a Client. Algorithms that match (or "de-duplicate") UIDs can be as simple as comparing two numbers, or as complicated as computing probabilistic matches.

Figure 15 (a) includes UID technologies already being considered. Source information includes biometrics, scan cards, question-and-answer, and the use of demographics and explicit identifiers. De-duplication instruments include directly matching (or linking) assigned, hashed, or encrypted values. Inconsistent hashing and distributed query are de-duplication instruments that do not simply match constructed UIDs. The UID technology termed "consent" merely checks whether Client permission was given. Each of these categories of UID technologies will be further described when they are assessed in Section 6. Figure 15 (b) shows some sample ways these are combined.

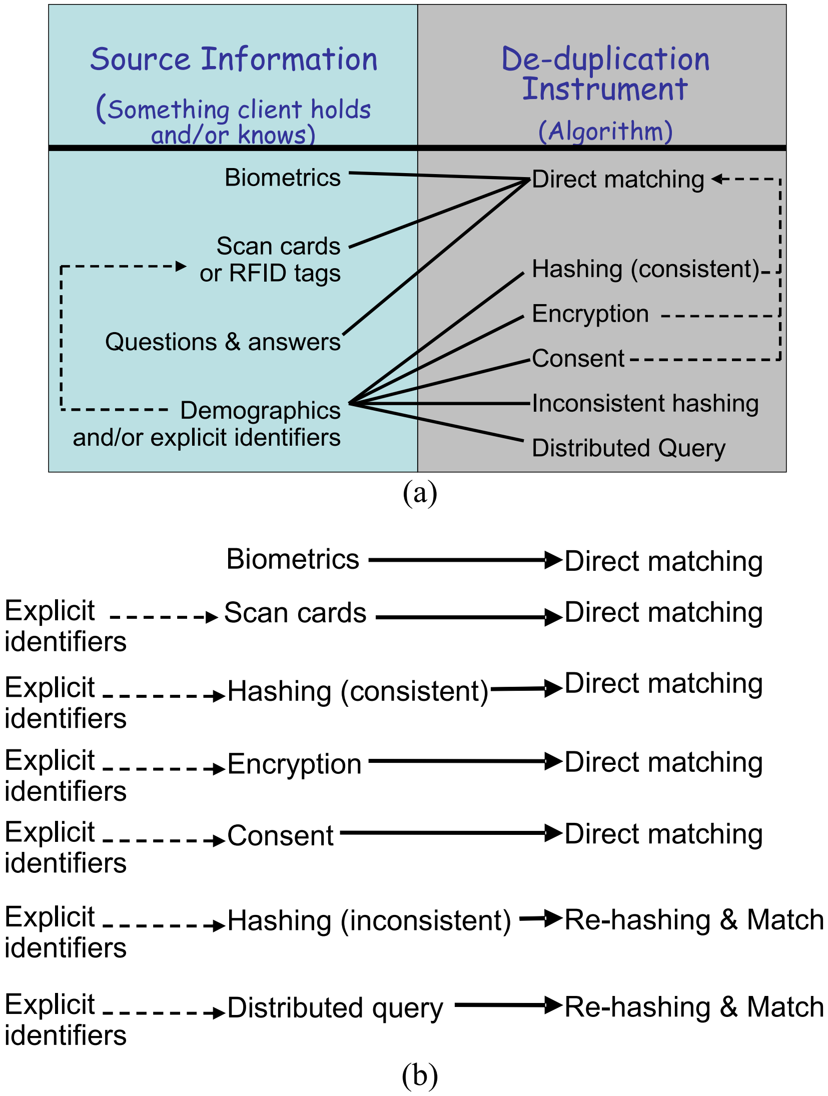


Figure 15. UID Technologies, assessed in Section 6, are broken down by source information and de-duplication instrument (a). Sample ways source and de-duplication instruments combine are shown in (b).

In Figure 15 (a), the solid line linkages between source information and de-duplication instruments show combinations of source information currently under consideration by some Planning Offices. Notice that biometrics, scan cards, question-and-answer, and demographic source information use direct matching to determine whether two UIDs match. Hashing, encryption, consent, inconsistent hashing, and distributed query all use demographics and/or explicit identifiers (e.g., Social Security number) as source information. The dashed lines in Figure 15 show secondary relationships. Demographics and explicit identifiers may be stored on scan cards. Hashed and encrypted values use direct matching for de-duplication. Consent also uses direct matching on demographics and/or explicit identifiers for de-duplication.

Assessing a UID technology involves producing Warranty and Compliance statements. Each of these are further described below.

5.2 Warranty statement (utility)

Given a Proposed Solution to the HMIS Unduplicated Count Problem, a Warranty shows that a reasonably accurate unduplicated accounting of records from Shelter Datasets is possible by the Planning Office. Below are fundamental issues that should be addressed by a Warranty.

The Warranty should demonstrate how de-duplication is done in the general case and identify the Proposed Solution's overall performance. Measures of accuracy should be included and cases that inflate or deflate the overall accounting should be addressed.

The behavior of the Proposed Solution using non-verifiable source information and verifiable source information should be examined. Particular attention should be given to the behavior of the Proposed Solution if Clients provide bad source information, such as purposeful name misspellings, wrong information, plausible differences in the information, or no information in part. Finally, consider the extent that the Proposed Solution can instill client confidence. This is particularly important when using non-verifiable source information because in these cases the system relies significantly on the cooperation of the Client.

Figure 16 lists considerations for Warranty statements.

WARRANTY (UTILITY) STATEMENT

Non-Verifiable source information	*If a UID is based on non-verifiable source information provided by the Client that is not truthful or is inconsistently used, what happens?*
Verifiable source information	*Can problems occur if the UID is based on verifiable source information? What if the information is not correct?*
Client confidence and trustworthiness	*The more Clients (and those who regularly intake Clients) trust the overall system and are encouraged to provide truthful information, the more likely Clients will actually provide truthful information. How trustworthy is the UID likely to be perceived by Clients (as well as by those who regularly intake Clients)? How would a lack of trust effect overall performance?*
Inflated accounting	*What are the circumstances under which de-duplication is likely to inflate the accounting? What are the circumstances in which a known Client is is not recognized (even if this does not actually inflate the count)? Explain the circumstances that generates these false negatives.*
Deflated accounting	*What are the circumstances under which de-duplication is likely to deflate the accounting? What are the circumstances in which a known Client is considered to be a different Client (even if this does not actually deflate the count)? Explain the circumstances that generates these false positives.*
Handling bad or missing input	*What is the effect of bad, incomplete, or missing source information on performance? How are these cases handled? (Note: "bad" information refers to accidental typing or other input mistakes.)*

Figure 16. Warranty Statements should seek to answer these questions.

5.3 Compliance statement (privacy)

Given a Proposed Solution to the HMIS Unduplicated Count Problem and publicly and readily available data and techniques, a Compliance Statement shows that the number of Clients who may be re-identified from the records in a Shelter Dataset is minimal. Below are fundamental issues that should be addressed by a Compliance Statement.

Consider any vulnerabilities the intimate stalker may exploit. Refer to Section 3.4.

Consider the ability to link Datasets, which include UIDs, to other available information in an attempt to re-identify Clients. Refer to Section 4.

“Dictionary attacks” should be considered. The idea of a dictionary attack is to generate UIDs for all possible source values and then see which results match UIDs stored in Dataset. Because the source that produced the UID is known, the information about the Client becomes known. Dictionary attacks assume the attacker has access to the UID technology and knowledge of what source information is used.

Here is an example of a dictionary attack. Assume a UID technology uses encryption to compute a number from the Client’s Social Security number. Even without knowing how encryption works (which will be discussed in Section 6.3), one can use a dictionary attack to learn the source information that generates a UID. Figure 17 shows an encryption method that when given a Social Security number produces a UID.

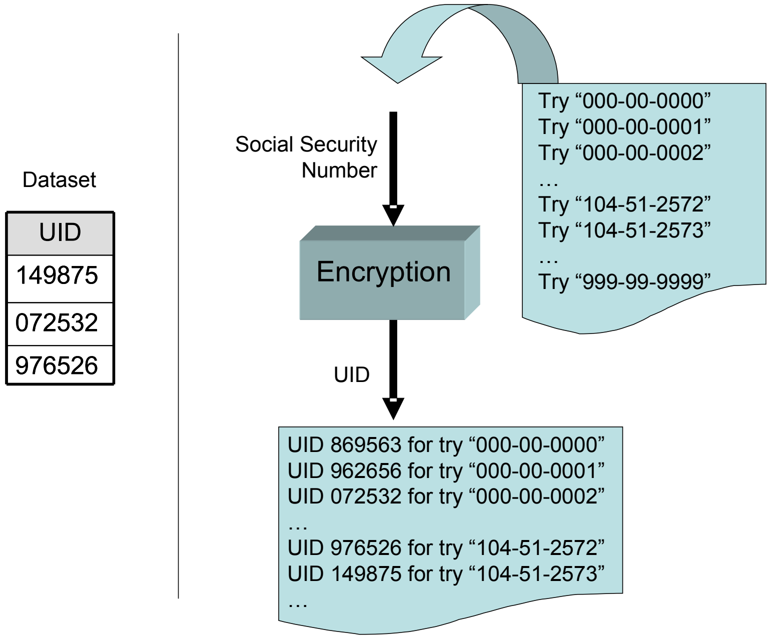


Figure 17. Example of a dictionary attack. Given a Dataset having UIDs 149875, 072532, and 976526, the knowledge that UIDs are encryptions of Social Security numbers, and access to the encryption function, a dictionary attack allows the UIDs to be learned by trying all possible Social Security numbers and seeing which Social Security numbers encrypt to the observed UIDs. In the example above, the Social Security number 104-51-2573 encrypts to 149875.

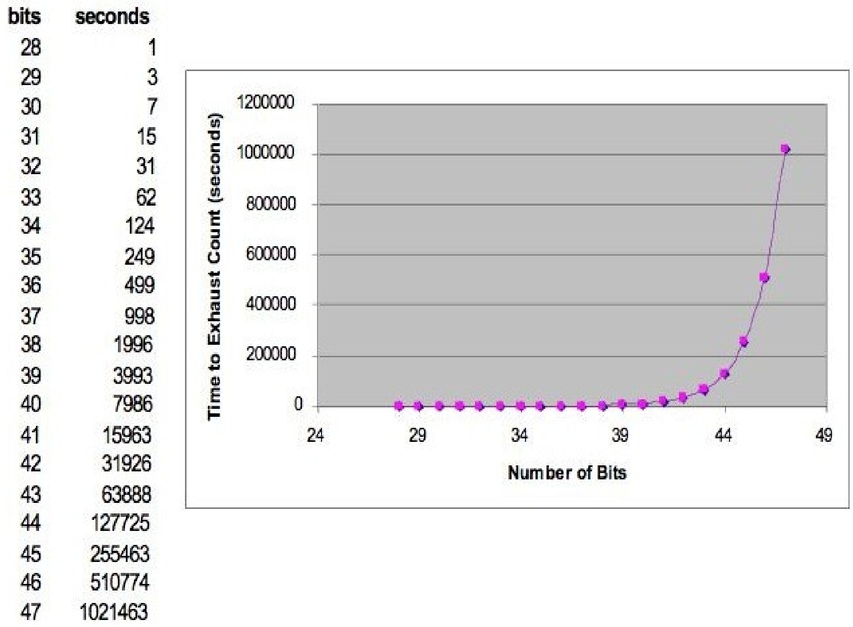
Suppose the Dataset contains the UIDs: 149875 and 072532. We can use a dictionary attack to learn the Clients' Social Security numbers that produced those UIDs by trying all possible 9-digit values and seeing which 9-digit Social Security numbers produce the UIDs that appear in the Dataset. As noted in Figure 17, the Social Security number "104-51-2572" produced UID 149875 and the Social Security number "000-00-0002" produced the UID 072532, so the Clients have the Social Security numbers "104-51-2572" and "000-00-0002," respectively.

A dictionary attack can be combined with linking to re-identify Clients by name. Assume a UID technology encrypts a combination of a Client's date of birth and gender to produce a UID. These same values also appear in the voter list (see Figure 10). So, computing UIDs for every voter in the voter list allows us to match UIDs in the Voter list to UIDs in the Dataset to re-identify Clients by name.

Of course, a dictionary attack can take a long time to compute. The example below reports the time needed for a dictionary attack to exhaust all possibilities using larger numbers on today's computers. To exhaust all possible 9-digit Social Security numbers (which requires 30 bits of storage) takes about 4 seconds. But to exhaust 36-bit numbers, which can store up to 11 digits, takes about 8 minutes. Using larger numbers requires more bits to store values; and, as the number of bits increases, the amount of time needed to test all possible values grows exponentially. Clearly, it is advantageous to use large numbers, as appropriate, in order to reduce the success of a dictionary attack.

Example (Exhausting Large Numbers)

> A simple program that counts from 0 to the largest integer that can be stored in x bits simulates a dictionary attack on x-bit numbers because it exhausts all possible values. Timing the execution of this program gives an estimate of the minimum time needed for a dictionary attack based on numbers requiring x-bits of storage. It ran under Java 1.5 on a 2003 vintage machine (Dell PrecisionTM 650 workstation having an Intel® Xeon™ Processor at 3.06 GHz, 512KB L2 cache, and 4GB RAM). Counting all possible Social Security numbers (0 to 999,999,999) took 4 seconds and used 30 bit numbers. Figure 18 shows the results of counting all integer values from 0 to the largest number that could be stored in 28 to 47 bits. Resulting times ranged from 1 second to 1021463 seconds (or 12 days), respectively.
>
> ■



bits	seconds
28	1
29	3
30	7
31	15
32	31
33	62
34	124
35	249
36	499
37	998
38	1996
39	3993
40	7986
41	15963
42	31926
43	63888
44	127725
45	255463
46	510774
47	1021463

Figure 18. Experimental results of time needed to exhaust 28 to 47 bit numbers. Time is the number of seconds needed to count from 0 to the maximum value stored in 28 to 47 bits on today's computers.

Figure 19 shows an equation that characterizes the values reported in Figure 18 and predicts the time needed for larger values which consume more bits. The variable x is the number of bits for the maximum value and y is the number of seconds needed to count from 0 to the maximum value. (The correlation coefficient R^2=0.9989 provides a measure of fitness based on a linear regression of the log of the actual and predicted values.) There are some interesting surprises. To exhaust 46-bit numbers capable of storing a concatenation of a typical person's Social Security number, month, day and year of birth, and gender in 15-digits takes about 6 days. To exhaust 64-bit numbers, which can store up to 20 digits, takes about 89 centuries!

$y=2.08^{x-28}$	28 bit	1 second
	30 bit	4 seconds
	35 bit	2.8 minutes
	40 bit	1.8 hours
	45 bit	3 days
	50 bit	3.8 months
	55 bit	12.3 years
	60 bit	4.8 centuries
	64 bit	89.4 centuries
	65 bit	186 centuries
	100 bit	25,220,489,437,291 centuries
	128 bit	20,301,442,123,378,100,000,000 centuries

Figure 19. Predicted time to exhaust x bit numbers. On left is an equation that characterizes the values reported in Figure 18. The variable x is the number of bits and y is the number of seconds needed to count from 0 to the maximum integer value that can be stored in x bits. On the right, are predictions of the time needed to exhaust all integers able to be stored in x bits.

One way to improve the wait time further is for multiple computers to work on different ranges of values at the same time thereby dividing the overall time across the machines. Figure 19 reports that exhausting 55-bit values would take one machine about 12.3 years. Conversely, if 3000 machines worked collectively in parallel, they would need no more than 36 minutes to exhaust all 55-bit values. While using thousands of machines may seem impossible, a single spammer reportedly used more than 2000 machines to send spam. The machines were idle on the Internet and physically located at different sites around the world, so no one realized that the spammer had compromised their operating systems and was running his own programs. Similarly, an attacker could co-opt thousands of machines to perform a dictionary attack.

The system using a UID technology should maintain client secrets even if the Planning Office mounts a dictionary attack. If the Planning Office has direct access to its own copy of the hash function, it can mount a dictionary attack in order to learn private information. Alternatively, the Planning Office can pad the values submitted to shelters with values of its own choosing in an attempt to learn private information. These kinds of accesses must not allow the Planning Office to learn private client information.

Beyond the intimate stalker threat, linking attacks, and dictionary attacks, an assessment should also examine to what extent the algorithm for producing the UID can be "reverse engineered." For example, given the following list of UIDs: 09121960ALIC, 10251974JANE, …, one can conclude that the UID is constructed by concatenating the month, day and year of birth with the first 4 letters of the first name. In this example, observing the UIDs revealed the method for constructing the UIDs. Given a Client's name and date of birth, the Client can be found in the Dataset.

A Compliance Statement should also identify any new legal or technical privacy risks that may be introduced based on the existence of the Proposed Solution's UID. This is considered "exposure."

Here is an example. If a Proposed Solution uses fingerprints as the source information for UIDs in such a way that a UID database is a fingerprint database, then the existence of the resulting database of Client fingerprints may be useful to law-enforcement. The existence of the database's usefulness to third parties poses new privacy concerns for Clients and thereby, increases exposure.

Figure 20 lists considerations for Compliance statements.

COMPLIANCE (PRIVACY) STATEMENT

Intimate Stalker	*What vulnerabilities exist for the intimate stalker?*
Re-identification: Linking	*What vulnerabilities exist for re-identification of UIDs (and Dataset) using data linkage on UIDs? What is the identifiability of the Dataset?*
Re-identification: Dictionary Attack	*What vulnerabilities exist for re-identification of UIDs (and Dataset) using a dictionary attack on UIDs?*
Re-identification: Reversal	*What is involved in reverse engineering the UID construction method?*
Exposure	*What legal or technical risks or liabilities may be introduced based on the existence of the resulting database or UID technology?*

System Trust

The overall system consists of intakers, who enter Client information, insiders, who have access to Client information for a variety of meritorious reasons, and the Shelters and Planning Offices themselves. Which parties are heavily trusted?

Figure 20. Compliance Statements should seek to answer these questions.

5.4 Other factors

There are many other factors that may contribute to a decision of which UID technology to use that are not part of the assessment. Among these are trust and economics. Where trust is placed differs among Proposed Solutions. Some solutions put more trust in the Shelters (e.g. distributed query), in the Clients (e.g., UID technologies using non-verifiable source information), or in the Planning Offices (e.g. consent).

Another key factor can be the economics of constructing, installing and maintaining the system. Some states are constructing systems for administrative oversight of social programs, so weaving HMIS requirements into those systems can be cost-effective, but doing so, may dictate the use of a particular UID technology.

Another factor can be available technical expertise.

While all these kinds of factors are important to the decision-making process, they are excluded from demonstrating the worthiness of the Proposed Solution. Warranty and Compliance statements demonstrate utility and privacy protection independent of these concerns.

5.5 Putting the pieces together

The goal of this section is to provide guidance on what should constitutes an assessment. The goal of the next section is to provide some overall assessments of 8 categories of technologies.

In summary, an assessment is a thorough review and analysis of a Proposed Solution that should be completed before a Proposed Solution is put to real-world use. Assessing a Proposed Technology requires a confluence of technology, policy, and sometimes law. Groups of people lacking the proper expertise or not focused on key issues pertinent to Warranty and Compliance issues can lead to poor assessments. The goal of this section and the next is to help those engaged in this process to ask themselves the right questions and to identify the right kinds of expertise needed. Along these lines, the following two recommendations are made.

Recommendation #11: *Given a Proposed Solution, a person skilled in statistical, computational and/or legal principles, as appropriate, should certify in writing that the Proposed Solution has a minimal risk of re-identification when the solution is considered with other publicly and readily available information and techniques. Such writing should address vulnerabilities for inappropriate re-identifications by various categories of insiders. This is termed a "Compliance Statement" and should be made available for inspection.*

Recommendation #12: *Given a Proposed Solution, a person skilled in statistical and/or computational principles, as appropriate, should certify in writing that the Proposed Solution provides a reasonably accurate unduplicated accounting of client visit patterns to shelters within the regional setting it is to be deployed. Such writing should include possible false match and missed match rates. This statement is termed a "Warranty" and should be made available for inspection.*

5.6 Privacy, not computer security

One word about computer security before continuing. This writing relates to data privacy concerns and not to computer security issues. It is assumed that any Proposed Solution operates in a computational environment having adequate computer security to authenticate users, limit access, combat intrusions and prevent eavesdropping. This writing does not address computer hacking, break-ins, viruses, or unauthorized computer users, because such issues appear to be adequately addressed with commercial computer security solutions. For general reference, see Pfleeger [21].

Instead, this writing addresses ways to limit authorized users from doing unauthorized tasks with available data. For example, the intimate stalker either has access to the Dataset already or obtains assistance from someone with access. Linking Dataset to other available information in order to re-identify Clients can only be done by someone with access to the Dataset. If someone does break-into the computer system and gains access to Dataset to attempt these things, the safeguards described in this writing will thwart their efforts. Described in this manner, these safeguards provide some privacy protection even in the face of a computer security breach. But more generally, these safeguards thwart unwanted activities by most of those who work with Dataset regularly.

6. Assessments of Initial UID Technologies

Assessments of 8 categories of initial UID technologies are presented in this section using the assessment criteria stated for Warranty and Compliance statements in the previous section. A summary of results appears at the end, in Section 6.9. (See Section 7 for VAWA compliance.)

The assessments presented in this section are not complete assessments. They examine only the UID technologies and not the accompanying policies or practices that may address noted concerns. Nonetheless, these assessments are useful in comparing UID technologies and in identifying the kinds of issues that accompanying policies and best practices need to address prior to use.

■	Most severe/difficult problem
▩	Moderate problem
▨	A problem
▒	May be a problem
□	No problem likely, or not applicable

Figure 21. Level of severity or difficulty of a problem is determined by shading.

For each of the UID technologies, the answers to the questions posed for Warranties (see Figure 16) and for Compliance statements (see Figure 20) are addressed with respect to that technology in the absence of accompanying policies or best practices. If a "problem" is described in answering the question, it should be addressed by accompanying policy or practice or by modification of the UID technology from the generally assumed form. A shaded code is assigned to denote the severity or difficulty of the problem: the darkest shading denotes a "serious problem," a dark hash pattern denotes a moderate problem, a light hash pattern denotes the existence of a "problem," a light shade with no pattern denotes a situation that "may be a problem," and no shading signals that there is not likely to be a problem. Figure 21 shows the shadings and patterns. Comments related to System Trust have no associated shading because these comments merely reflect where trust is placed.

The following categories of UID technologies are examined in the noted subsections.

6.1. Encoding
6.2. Hashing
6.3. Encryption
6.4. Scan Cards / RFID
6.5. Biometrics
6.6. Consent
6.7. Inconsistent hashing
6.8. Distributed query

Section 6.9 provides a comparative summary.

6.1 Encoding

Using "encoding" to produce UIDs simply involves concatenating parts of source information to form a UID. De-duplication is then performed by simply matching resulting UID values.

Figure 22 provides an example of a UID constructed by encoding the fileds {*date of birth*, *gender*, *ZIP*}. Specifically:

encode(9/12/1960, F, 37213) = "09121960F37213"

In this example, the digits of the date of birth, a letter for gender, and the 5-digits residential ZIP code are merely concatenated. While this example uses all characters in the source information, encoding sometimes uses only some characters, such as using the first 5 letters of a person's last name.

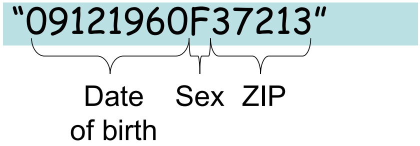


Figure 22. Example of making a UID by encoding {date of birth, gender, ZIP}.

An obvious problem with encoding is that given a series of UIDs and some source information, an attacker can often deduce what parts of which source information appears in the UID and where in the UID it appears.

Figure 23 and Figure 24 provide a gross assessment of encoding as a UID technology. Issues related to utility and the warranty statement appear in Figure 23. Issues related to privacy and the compliance statement appear in Figure 24. While shadings may identify some problems as being of severe or moderate concern, these problems may be sufficiently addressed with straightforward practices, policies, or technology decisions.

(Additionally, encoding is problematical under VAWA; see Section 7.2.4.)

ENCODING --WARRANTY (UTILITY) STATEMENT

Non-Verifiable source information	■	*If a UID is based on non-verifiable source information provided by the Client that is not truthful or is inconsistently used, what happens?* Serious de-duplication problems are likely if Clients provide non-verifiable source information inconsistently. On the other hand, source information that is not truthful, but consistently provided, is typically not a problem.
Verifiable source information	░	*Can problems occur if the UID is based on verifiable source information?* Using invariant Client information that can be consistently verified on each visit is likely to avoid problems. Even if the information is not correct, but consistently verified on each visit, no problems are likely. An example of invariant verifiable Client information is a reliably captured biometric, but biometrics seem unlikely source information for encoding (refer to hashing, encryption, or inconsistent hashing). So, determining what would constitute verifiable Client information for encoding would be important.
Client confidence and trustworthiness	▓	*How trustworthy is the UID likely to be perceived by Clients (as well as by those who regularly intake Clients)?* Encoded UIDs tend to be transparent, which can limit Client and intaker confidence by exposing information. Accompanying practices should seek to build Client and intaker trust. An example of a transparent code that would still maintain trust would be to allow Clients to make up their own UID or to use answers to simple questions as source information (see Section 5.1.1).
Inflated accounting	░	*What are the circumstances under which de-duplication is likely to inflate the accounting?* Count inflation can occur in cases where a Client provides different source information on different visits. In these cases, different UIDs are generated and therefore will not match to each other even though they are assigned to the same Client. This relates to the comment above on non-verifiable information.
Deflated accounting	■	*What are the circumstances under which de-duplication is likely to deflate the accounting?* Count deflation can occur in cases in which a Client provides incomplete or missing information or different source information on different visits, or in which a bad method is used for generating UIDs. In these cases, the same UID is generated for different Clients and therefore visit information will combine inappropriately, generating serious accounting problems. Deflation is more likely than inflation.

…continued on next page …

Handling bad or missing input	■	*What is the effect of bad, incomplete, or missing source information on performance?* Typing mistakes that go uncorrected, as well as incomplete or missing information, can generate different UIDs for a Client than would have been generated with complete and properly entered information. This tends to inflate accounting by generating spurious UIDs for Clients having multiple visits. On the other hand, having the same incomplete and missing information across Clients will deflate accounting because different Clients would have the same UID. See comments on inflated and deflated accounting above.

■	Most severe/difficult problem
▓	Moderate problem
▓	A problem
░	May be a problem
	No problem likely, or not applicable

Figure 23. Gross Warranty assessment of encoding as a UID technology.

ENCODING –COMPLIANCE (PRIVACY) STATEMENT

Intimate Stalker	■	*What vulnerabilities exist for the intimate stalker?* In typical cases where demographics are the source information encoded, serious problems may exist. Demographics tend to be visible within the encoding, making identification more transparent to an intimate stalker.
Re-identification: Linking	■	*What vulnerabilities exist for re-identification of UIDs (and Dataset) using data linkage on UIDs?* Because demographics tend to be the source information used with encoding and demographics appear in other available data, linking tends to be a serious problem. Analysis of specific risk should be based on the re-identification of demographics over the actual population from which Clients are drawn.
Re-identification: Dictionary Attack	■	*What vulnerabilities exist for re-identification of UIDs (and Dataset) using a dictionary attack on UIDs?* A dictionary attack can be done by executing the encoding function over all legal combinations of source information. For any generated UID that matches a UID in the Dataset, the Client's source information is learned. This may pose a serious problem depending on the source information and encoding method used. A combination dictionary-attack and linking attack can also be a problem. For example, suppose some other data (Other Data) is to be linked to a Dataset in which UIDs are encoded using source information and the same source information appears in Other Data. UIDs can be produced for the source information in Other Data, and then, UIDs in Dataset are matched to UIDs in Other Data to link Client data.
Re-identification: Reversal	■	*What is involved in reverse engineering the UID construction method?* Because encodings tend to be transparent, casual (or visual) inspection can often be used to describe the encoding algorithm. Even in cases where the encoding appears more cryptic, inspecting known cases can often reveal the encoding method.
Exposure	▨	*What legal or technical risks or liabilities may be introduced based on the existence of the resulting database or UID technology?* The existence of encodings enable risks of linking described above and can make demographics on Clients transparent which can increase re-identification risks beyond the HMIS context.

■	Most severe/difficult problem
▨	Moderate problem
▨	A problem
░	May be a problem
□	No problem likely, or not applicable

…continued on next page …

System Trust
Which parties are heavily trusted?

All insiders are heavily trusted not to decode UIDs or exploit the knowledge they may learn about the encoding scheme. If the encoding scheme is obscure, then the scheme itself is heavily trusted in the belief that no one, no matter how heavily motivated, will learn or share the scheme. Additionally, if the encoding scheme is obscure, insiders with access to the encoding method are heavily trusted.

Figure 24. Gross Compliance assessment of encoding as a UID technology.

6.2 Hashing

Using "hashing" to produce UIDs involves computing a number from source information. De-duplication is then performed by simply matching UID values.

Figure 25 provides an example of making a UID by hashing the fields {*date of birth*, *gender*, *ZIP*}. Specifically:

hash(9/12/1960, F, 37213) = "8126r1329ws"

Unlike encoding, the hashed value is not transparent, as it was with encoding (Section 6.1).

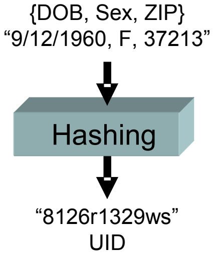


Figure 25. Example of making a UID by hashing {date of birth, gender, ZIP}.

Hashed UIDs are consistently produced. That is, each time the hash function is given the same input, it produces the same UID.

A vendor can create their own hash function, but it has been shown that these "ad hoc" approaches can be reversed, especially if someone is highly motivated to do so. Protection using an ad hoc hash function is good only as long as no one learns the actual hash function used. Rather than using ad hoc hash functions, cryptographically "strong" hash methods are highly recommended. With a strong hash function, everyone can examine the method being used, but even with intense inspection, it has been proven that no one can reverse the process without performing more computation than can be reasonably performed [22].

Hash functions have the property that they do not preserve the natural ordering typically found in source values. Two consecutive values (e.g. ZIP codes 37212 and 37213) tend to have radically different hashed values (e.g., "x41768" and "z1Rx5G"). This is good for privacy, but can be bad for utility.

Figure 26 and Figure 27 provide a gross assessment of hashing as a UID technology. Issues related to utility and the warranty statement appear in Figure 26. Issues related to privacy and the compliance statement appear in Figure 26. While shadings may identify some problems as being of severe or moderate concern, these problems may be sufficiently addressed with straightforward practices, policies, or technology decisions.

(Additionally, hashing is problematical under VAWA; see Section 7.2.5.)

HASHING –WARRANTY (UTILITY) STATEMENT

Non-Verifiable source information	■	*If a UID is based on non-verifiable source information provided by the Client that is not truthful or is inconsistently used, what happens?* Serious de-duplication problems are likely if Clients provide non-verifiable source information inconsistently because similar source values have radically different hashed values. On the other hand, source information that is not truthful, but consistently provided, is typically not a problem.
Verifiable source information	□	*Can problems occur if the UID is based on verifiable source information?* Using invariant Client information that can be consistently verified on each visit is likely to avoid problems. Even if the information is not correct, but consistently verified on each visit, no problems are likely. An example of invariant verifiable Client information is a reliably captured biometric.
Client confidence and trustworthiness	▒	*How trustworthy is the UID likely to be perceived by Clients (as well as by those who regularly intake Clients)?* Hashed UIDs tend to appear cryptic, which can instill Client and intaker confidence. However, problems can emerge in cases where the requested source information is sensitive, notwithstanding the cryptic appearance of the UID itself. Educating Clients and those who perform intake regularly and/or issuing privacy notices may help.
Inflated accounting	▒	*What are the circumstances under which de-duplication is likely to inflate the accounting?* Count inflation can occur in cases where a Client provides different source information on different visits (see comments for non-verifiable source information above). In these cases, different UIDs are generated and therefore will not match to each other even though they are assigned to the same Client.
Deflated accounting	■	*What are the circumstances under which de-duplication is likely to deflate the accounting?* Count deflation can occur in cases in which a Client provides incomplete or missing information or different source information on different visits, or in which a bad method is used for generating UIDs. In these cases, the same UID is generated for different Clients and therefore visit information will combine inappropriately, generating serious accounting problems. Deflation is more likely than inflation.

…continued on next page …

Handling bad or missing input	■	*What is the effect of bad, incomplete, or missing source information on performance?* Typing mistakes that are go uncorrected, as well as incomplete or missing information, can generate different UIDs for a Client than would have been generated with complete and properly entered information. This tends to inflate accounting by generating spurious UIDs for Clients having multiple visits. On the other hand, incomplete and missing information is likely to deflate accounting because different Clients whose entries are missing the same information may have the same UID.

■	Most severe/difficult problem
▓	Moderate problem
▓	A problem
░	May be a problem
□	No problem likely, or not applicable

Figure 26. Gross Warranty assessment of hashing as a UID technology.

HASHING –COMPLIANCE (PRIVACY) STATEMENT

Intimate Stalker	■	*What vulnerabilities exist for the intimate stalker?* In typical cases where demographics is the source information used with hashing, serious problems may exist. Access to the hash function can allow the intimate stalker (working with a compromised insider) to generate a Client's UID, and then to use the UID to identify the Client's Shelter location in the Dataset. Control and auditing of hash function use is important to thwarting this problem.
Re-identification: Linking	■	*What vulnerabilities exist for re-identification of UIDs (and Dataset) using data linkage on UIDs?* Because demographics tend to be the source information used with hashing and demographics appear in other available data, linking tends to be a problem if access to the hash function is not controlled and audited. Practices should limit and account for hash function use. Risk analysis should be based on the re-identification of demographics over the actual population from which Clients are drawn.
Re-identification: Dictionary Attack	■	*What vulnerabilities exist for re-identification of UIDs (and Dataset) using a dictionary attack on UIDs?* A dictionary attack can be done by executing the hash function over all legal combinations of source information. For any generated UID that matches a UID in Dataset, the Client's source information is learned. This may pose a serious problem depending on source information and hash method used. A combination dictionary-attack and linking attack can also be a problem. For example, suppose some other data (Other Data) is to be linked to a Dataset in which UIDs are hashed using source information and the same source information appears in Other Data. UIDs can be produced for the source information in Other Data, and then, the UIDs in Dataset are matched to the UIDs in Other Data to link Client data to Other Data. Practices should limit and account for uses of the hash function.
Re-identification: Reversal	▓	*What is involved in reverse engineering the UID construction method?* If a "strong" hash function is used, then it is highly unlikely that the method will be reversed. For this reason, strong rather than ad hoc hash functions should be used. If strong methods are not used, then attention must be paid to the ability to reverse the method.
Exposure	□	*What legal or technical risks or liabilities may be introduced based on the existence of the resulting database or UID technology?* The existence of hashed UIDs used only in the HMIS-context is not likely to expose Clients to additional risks beyond those mentioned above.

■	Most severe/difficult problem
▓	Moderate problem
▓	A problem
░	May be a problem
□	No problem likely, or not applicable

…continued on next page …

System Trust
Which parties are heavily trusted?

If the hash function is ad hoc (not strong), then the function itself is heavily trusted in the belief that no one, no matter how heavily motivated, will reverse the function. It also requires trusting the developer of the ad hoc hash function.

Additionally, no matter whether the hash function is ad hoc or strong, insiders with access to the hash function are heavily trusted.

Figure 27. Gross Compliance assessment of hashing as a UID technology.

6.3 Encryption

Using encryption to produce a UID involves computing a number from source information. De-duplication is then performed by simply matching UID values. This is the same as hashing (Section 6.2), except with encryption there exists a "key" such that whoever has the key can reverse the process to take a UID and reveal some (or all) of the source information that produced it.

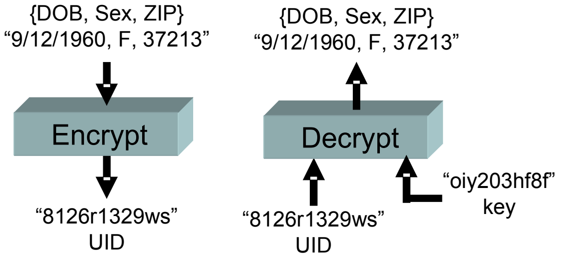


Figure 28. Example of making a UID by encrypting {date of birth, gender, ZIP}. With the key, the process is reversed to reveal the original source information.

Figure 28 provides an example of making a UID by encryption the fields {*date of birth*, *gender*, *ZIP*}. Specifically:

encrypt(9/12/1960, F, 37213) = "8126r1329ws"

Then,

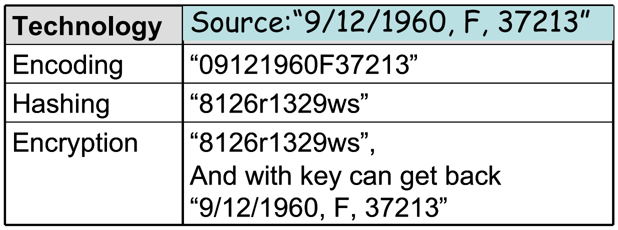
decrypt(*key*, "8126r1329ws") = "9/12/1960, F, 37213"

Encrypted UIDs, as with hashing, are consistently produced. Each time the encryption function is given the same input, it produces the same UID.

A vendor can create their own encryption function, but it has been shown that these "ad hoc" approaches can be reversed, especially if someone is highly motivated to do so. [This is the same as was discussed with hashing in Section 6.2.] Protection using an ad hoc encryption function is good only as long as no one learns the actual encryption function used. Rather than using ad hoc encryption functions, cryptographically "strong" encryption methods are highly recommended. With a strong encryption function, everyone can examine the method being used, but even with intense inspection, it has been proven that no one can reverse the process without the key [22].

Encryption functions have the property that they do not preserve the natural ordering typically found in source values. [This is the same as was discussed with hashing in Section 6.2.] Two consecutive values (e.g. ZIP codes 37212 and 37213) tend to have radically different encrypted values (e.g., "x41768" and "z1Rx5G"). This is good for privacy, but can be bad for utility.

Encoding, hashing and encryption are very similar, as shown in Figure 29. However, encoding tends to visibly reveal source information where as hashing and encryption values do not. Encryption, in comparison to hashing, has a key that can reverse the process.



Technology	Source:“9/12/1960, F, 37213”
Encoding	“09121960F37213”
Hashing	“8126r1329ws”
Encryption	“8126r1329ws”, And with key can get back “9/12/1960, F, 37213”

Figure 29. Comparison of encoding, hashing, and encryption. Encoding tends to transparently reveals the original source values. Encryption has a key that can reverse the process.

See Figure 30 and Figure 31 for a gross assessment of encryption as a UID technology. Issues related to utility and the warranty statement appear in Figure 30. Issues related to privacy and the compliance statement appear in Figure 31. While shadings may identify some problems as being of severe or moderate concern, these problems may be sufficiently addressed with straightforward practices, policies, or technology decisions.

(Additionally, encryption is problematical under VAWA; see Section 7.2.6.)

ENCRYPTION –WARRANTY (UTILITY) STATEMENT

Non-Verifiable source information	■	*If a UID is based on non-verifiable source information provided by the Client that is not truthful or is inconsistently used, what happens?* Serious de-duplication problems are likely if Clients provide non-verifiable source information inconsistently. On the other hand, source information that is not truthful, but consistently provided, is typically not a problem.
Verifiable source information	□	*Can problems occur if the UID is based on verifiable source information?* Using invariant Client information that can be consistently verified on each visit is likely to avoid problems. Even if the information is not correct, but consistently verified on each visit, no problems are likely. An example of invariant verifiable Client information is a reliably captured biometric.
Client confidence and trustworthiness	▓	*How trustworthy is the UID likely to be perceived by Clients (as well as by those who regularly intake Clients)?* Encrypted UIDs tend to appear cryptic, which can instill Client and intaker confidence. However, problems can emerge in cases where the requested source information is sensitive, notwithstanding the cryptic appearance of the UID itself. Educating Clients and those who perform intake regularly and/or issuing privacy notices may help. The existence of a key that can unlock Client information may also reduce Client confidence.
Inflated accounting	░	*What are the circumstances under which de-duplication is likely to inflate the accounting?* Count inflation can occur in cases where a Client provides different source information on different visits (see comments for non-verifiable source information above). In these cases, different UIDs are generated and therefore will not match to each other even though they are assigned to the same Client.
Deflated accounting	■	*What are the circumstances under which de-duplication is likely to deflate the accounting?* Count deflation can occur in cases in which a Client provides incomplete or missing information or different source information on different visits, or in which a bad method is used for generating UIDs. In these cases, the same UID is generated for different Clients and therefore visit information will combine inappropriately, generating serious accounting problems. Deflation is more likely than inflation.

…continued on next page …

Handling bad or missing input	■	*What is the effect of bad, incomplete, or missing source information on performance?* Typing mistakes that are go uncorrected, as well as incomplete or missing information, can generate different UIDs for a Client than would have been generated with complete and properly entered information. This tends to inflate accounting by generating spurious UIDs for Clients having multiple visits. On the other hand, incomplete and missing information is likely to deflate accounting because different Clients whose entries are missing the same information may have the same UID.

■	Most severe/difficult problem
▓	Moderate problem
▓	A problem
░	May be a problem
□	No problem likely, or not applicable

Figure 30. Gross Warranty assessment of encryption as a UID technology.

ENCRYPTION –COMPLIANCE (PRIVACY) STATEMENT

Intimate Stalker	■	*What vulnerabilities exist for the intimate stalker?* In typical cases where demographics is the source information used with encryption, serious problems may exist. Access to the encryption function, or the key with the decryption function, can allow the intimate stalker (working with a compromised insider) to generate a Client's UID, and then to use the UID to identify the Client's Shelter location in the Dataset. Control and auditing of the encryption and decryption functions are important to thwarting this problem.
Re-identification: Linking	■	*What vulnerabilities exist for re-identification of UIDs (and Dataset) using data linkage on UIDs?* Because demographics tend to be the source information used with encryption and demographics appear in other available data, linking tends to be a problem if access to the encryption and decryption functions are not controlled and audited. Practices should limit and account for encryption and decryption use. Risk analysis should be based on the re-identification of demographics over the actual population from which Clients are drawn.
Re-identification: Dictionary Attack	■	*What vulnerabilities exist for re-identification of UIDs (and Dataset) using a dictionary attack on UIDs?* A dictionary attack can be done by executing the hash function over all legal combinations of source information. For any generated UID that matches a UID in Dataset, the Client's source information is learned. This may pose a serious problem depending on source information and encryption method used. A combination dictionary-attack and linking attack can also be a problem. For example, suppose some other data (Other Data) is to be linked to a Dataset in which UIDs are encrypted using source information and the same source information appears in Other Data. UIDs can be produced for the source information in Other Data, and then, the UIDs in Dataset are matched to the UIDs in Other Data to link Client data to Other Data. Practices should limit and account for uses of the encryption function and also for key use.
Re-identification: Reversal	▒	*What is involved in reverse engineering the UID construction method?* If a "strong" encryption function is used, then it is highly unlikely that the method will be reversed. For this reason, strong rather than ad hoc encryption functions should be used.

…continued on next page …

Exposure	▩	*What legal or technical risks or liabilities may be introduced based on the existence of the resulting database or UID technology?* The existence of encrypted UIDs means there exists a key that can unlock the UIDs without permission, thereby increasing Client risks beyond the HMIS context.

■	Most severe/difficult problem
▩	Moderate problem
▩	A problem
▒	May be a problem
□	No problem likely, or not applicable

System Trust

Which parties are heavily trusted?

If the encryption function is ad hoc (not strong), then the function itself is heavily trusted in the belief that no one, no matter how heavily motivated, will reverse the function. It also requires trusting the developer of the ad hoc encryption function.

Any party that has access to the decryption key is heavily trusted.

Additionally, no matter whether the encryption function is ad hoc or strong, insiders with access to the encryption function are heavily trusted.

Figure 31. Gross Compliance assessment of encryption as a UID technology.

6.4 Scan Cards/RFID

Using Scan Cards as a UID technology involves issuing a card containing a UID to each Client who presents for service. The card can store a photo, serial#, randomly assigned number, and/or demographics. Figure 32 shows a depiction of a scan card in which only a serial number and picture appear.

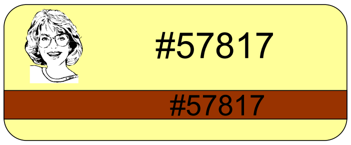


Figure 32. Depiction of a scan card with a serial number and photograph visible. The magnetic strip stores the serial number, but the serial number stored on the strip is not visible to the naked eye.

Scan cards that have a magnetic strip on one side resemble credit cards. Information is stored on the magnetic strip that can be read by a card reader even though the information is not visible to the human eye. In fact, these magnetic strips are typically readable by most card readers, and therefore, the ability to read scan cards is not limited to card authorized readers. Card readers outside those located at Shelters could read the cards.

Radio frequency identification (RFID) cards have no magnetic strip. Information is still stored within the card and can be read by an RFID reader. But unlike magnetic strip cards, RFID content intended for one reader is not as easily read by other readers. In fact, expensive RFID cards and readers offer exclusive protection. Only authorized readers are easily able to read specific kinds of cards. Finally, RFID cards come in a variety of sizes, some smaller than a dime (and many cost less than a dime too).

The decision of what appears printed on the card is important in assessing its use as a UID technology. If Shelter information appears, others may learn information about the Client from merely viewing the card.

The information stored on the card is the UID. The source information can be a randomly assigned number, demographics, or some other value. If a serial or random number is assigned, the Planning Office will most likely have to coordinate issuances of numbers across Shelters.

See Figure 33 and Figure 34 for a gross assessment of using scan cards as a UID technology. Issues related to utility and the warranty statement appear in Figure 33. Issues related to privacy and the compliance statement appear in Figure 34. While shadings may identify some problems as being of severe or moderate concern, these problems may be sufficiently addressed with straightforward practices, policies, or technology decisions.

(Additionally, scan cards / RFID may be okay under VAWA; see Section 7.2.3.)

SCAN CARDS / RFIDs –WARRANTY (UTILITY) STATEMENT

Non-Verifiable source information	▓	*If a UID is based on non-verifiable source information provided by the Client that is not truthful or is inconsistently used, what happens?* Assume non-verifiable information is the basis for a UID stored on a card. Then, if the Client consistently uses the card, no problem is likely. But if cards are borrowed or swapped, or if Clients have multiple cards issued with different UIDs (e.g., with card replacement), problems are likely.
Verifiable source information	░	*Can problems occur if the UID is based on verifiable source information?* Using invariant Client information that can be consistently verified on each visit is likely to avoid problems. Even if the information is not correct, but consistently verified on each visit, no problems are likely. An example of invariant verifiable Client information that can be stored on a scan card is a reliably captured biometric (see Section 6.5). Printing photographs on the card may be considered a means to verify identity, but intake personnel must be trained to actually verify appearance.
Client confidence and trustworthiness	█	*How trustworthy is the UID likely to be perceived by Clients (as well as by those who regularly intake Clients)?* Scan cards may pose serious problems based on the existence of the card and on information appearing on the card. Assume a Client was issued a card and subsequently returned home to the abuser. The card, if found, can instigate trouble. Further, if information about the location of the Shelter or the UID itself are actually printed on the card, the intimate stalker may gain sensitive information.
Inflated accounting	█	*What are the circumstances under which de-duplication is likely to inflate the accounting?* The issuance of additional scan cards to the same person can inflate the count if new cards have different UIDs. Accompanying practices should address how registration of cards is done and how lost cards are handled. This is likely to be a common problem. Swapping cards among Clients does not actually inflate the count, but it does generate false visit patterns in which visits of one Client are incorrectly associated with another.
Deflated accounting	░	*What are the circumstances under which de-duplication is likely to deflate the accounting?* Count deflation is not likely to occur with scan cards unless the information used to generate the UID associated with the card is badly chosen. Most ways in which UIDs stored on scan cards are likely to be generated pose no problem. For example, randomly generated UIDs would not pose a problem. But if source information produces the same UIDs for different people (i.e., different cards assigned to different Clients but having the same UIDs), then visit information would combine inappropriately, generating accounting problems.

…continued on next page …

Handling bad or missing input	░	*What is the effect of bad, incomplete, or missing source information on performance?* Bad or missing information is not likely to effect the performance with scan cards unless the information used to generate the UID associated with the card is badly chosen. Most ways in which UIDs stored on scan cards are likely to be generated pose no problem. For example, randomly generated UIDs would not pose a problem. But if the method relied on source information that could have bad or missing information, then deflated accounting is possible because different Clients whose entries are missing the same information may have the same UID.

■	Most severe/difficult problem
▓	Moderate problem
▓	A problem
░	May be a problem
□	No problem likely, or not applicable

Figure 33. Gross Warranty assessment of using scan cards as a UID technology.

SCAN CARDS / RFIDs –COMPLIANCE (PRIVACY) STATEMENT

Intimate Stalker	▓	*What vulnerabilities exist for the intimate stalker?* In cases where printable information appearing on the card itself includes Shelter location or the UID itself, viewing the card may reveal sensitive information. Practices should address information appearing on the card and its possible use by the stalker. Care nust also be taken that the UID dies not reveal or use information available to the intimate stalker.
Re-identification: Linking	▓	*What vulnerabilities exist for re-identification of UIDs (and Dataset) using data linkage on UIDs?* If demographics are stored or printed on the card, linking will be a problem. Risk analysis should be based on demographics over the actual population from which Clients are drawn. However, other possibilities, beyond demographics, exist as the basis for providing UIDs for scan cards.
Re-identification: Dictionary Attack	░	*What vulnerabilities exist for re-identification of UIDs (and Dataset) using a dictionary attack on UIDs?* If the UID associated with a Scan Card is just a random number, then a dictionary attack is not likely. However, if the UID associated with a Scan Card uses demographics or biometrics, then vulnerabilities may exist (see Section 5.3 and Section 6.5).
Re-identification: Reversal	░	*What is involved in reverse engineering the UID construction method?* If the UID associated with a Scan Card is just a random number, then reversal is not likely. However, if the UID associated with a Scan Card uses encoding or hashing, then vulnerabilities may exist (see Section 6.1 and Section 6.2)).
Exposure	█	*What legal or technical risks or liabilities may be introduced based on the existence of the resulting database or UID technology?* The existence of the Scan Card in the Client's possession and any information printed on the card can expose a Client's consumption of Shelter services to an intimate abuser, for example. Care should be taken about the information printed on the card. The severity of this problem can be easily resolved by avoiding such printing on the card.

█	Most severe/difficult problem
▓	Moderate problem
▓	A problem
░	May be a problem
□	No problem likely, or not applicable

System Trust
Which parties are heavily trusted?

Assuming a scan card stores only a randomly assigned number and no printed information is visible, then scan cards place trust in Clients in the belief that Clients will use the same card on recurring visits, will not swap cards and will provide the same source information on card replacement or re-issuance.

Figure 34. Gross Warranty assessment of using scan cards as a UID technology.

6.5 Biometrics

Using a biometric as source information for a UID technology has the advantage that the biometric is something always present with the Client and that typically does not change. The most common biometric is a fingerprint. Figure 35 shows how a fingerprint is used as source information. A fingerprint can be used as source information to a hash or encryption function or the fingerprint itself can be the UID.

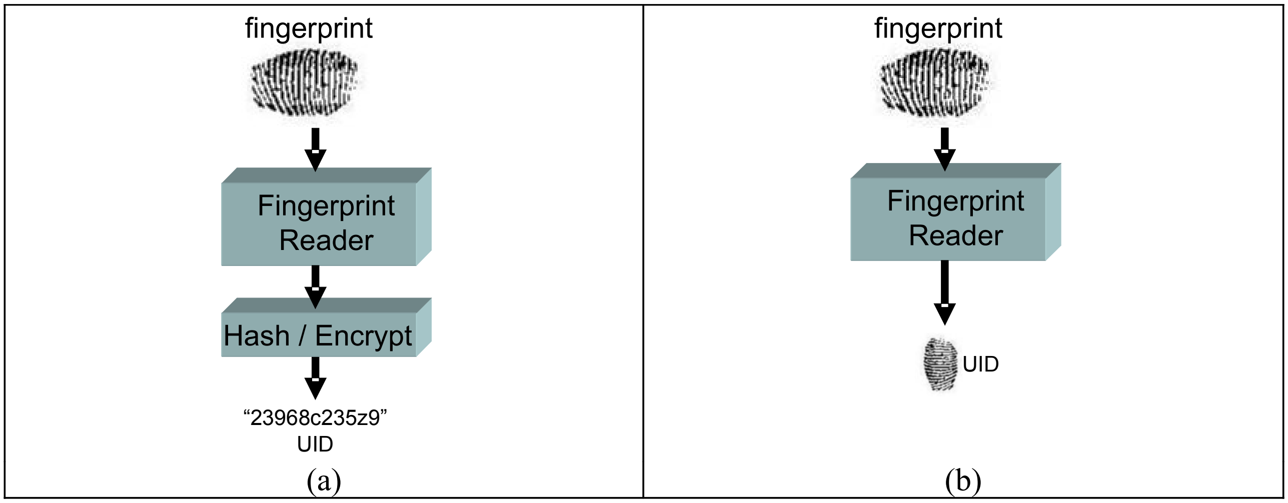


Figure 35. Fingerprint as source information to a hash or encryption function to generate a UID (a); or, used as the UID iteself (b).

Fingerprint readers have become inexpensive and as a result, fingerprint reading is becoming popular for all kinds of new uses, such as a way to gain access to a car or a refrigerator or to use a computer keyboard. Of course, inexpensive capture devices tend to be horribly inaccurate, but reasonably priced devices perform reasonably well. It is important to test the accuracy of a fingerprint system on the population with which it will be used. The combination of a particular fingerprint system with a specific population should be checked for consistency and accuracy. Check that the same person is recognized to be the same person (and not someone else). Also confirm that a person who has been in the system continues to be recognized (and not considered a new person).

For some explained and unexplained reasons, there are some people whose fingerprints cannot be reliably captured [23]. Finger cuts, scars, amputations, disease, infection, and overall disabilities and abnormalities can pose fingerprint capture problems. Hands having excessive moisture or dryness can frustrate fingerprint capture. Unofficial FBI statements claim that persons involved with certain drugs and persons who regularly scrape their fingertips on abrasive surfaces, such as concrete, cannot be reliably fingerprinted. If so, some homeless people who spend significant time on concrete sidewalks may be difficult to fingerprint.

If fingerprint images are captured and used as UIDs, Shelters and Planning Offices would maintain a de facto fingerprint database of Clients. The existence of such a database may invite linking requests (unofficial and official), especially from law enforcement. Whether matching latent prints to a crime scene or confirming identity, law enforcement requests serviced by Shelters may alter how some Shelters and Clients have historically viewed the homeless service environment. An increase in court orders demanding copies of Client prints, the UID construction method, and all Client UIDs is a likely possibility.

See Figure 36 and Figure 37 for a gross assessment of using biometrics in UID technologies. Issues related to utility and the warranty statement appear in Figure 36. Issues related to privacy and the compliance statement appear in Figure 37. While shadings may identify some problems as being of severe or moderate concern, these problems may be sufficiently addressed with straightforward practices, policies, or technology decisions.

(Additionally, biometrics is not allowed under VAWA; see Section 7.2.2.)

BIOMETRICS (fingerprints) –WARRANTY (UTILITY) STATEMENT

Non-Verifiable source information		*If a UID is based on non-verifiable source information provided by the Client that is not truthful or is inconsistently used, what happens?* Does not require non-verifiable source information from Clients.
Verifiable source information		*Can problems occur if the UID is based on verifiable source information?* A biometric that can be consistently and reliably captured can provide independent, invariant Client information that is not likely to be bad or to cause problems.
Client confidence and trustworthiness	░	*How trustworthy is the UID likely to be perceived by Clients (as well as by those who regularly intake Clients)?* UIDs based on biometrics are generally invariant to Client trust though some attention should be given to establishing Client acceptance of what may be perceived as an invasive process. Otherwise, Clients may purposefully try to generate bad captures, if possible, in an attempt to thwart the system.
Inflated accounting	▓	*What are the circumstances under which de-duplication is likely to inflate the accounting?* Inexpensive technology or poor quality biometrics can inflate counts when the same person generates different UIDs. In most cases, Clients are likely to undergo a registration process to generate a database of known Clients. Then, when a Client appears on a subsequent visit, if the presenting biometric is not found, the count is not inflated, but administering the process is slowed by having to repeat captures until a matching biometric is found. Attention should be spent on testing the accuracy of the biometric capture on the specific Client population. Sometimes, using multiple captures can improve results. Another possible remedy is to use better technology.
Deflated accounting	▓	*What are the circumstances under which de-duplication is likely to deflate the accounting?* Inexpensive technology or poor quality biometrics can deflate counts when multiple people map to the same UID. Attention should be spent on testing the accuracy of the biometric capture on the specific Client population. Sometimes, using multiple captures can improve results. Another possible remedy is to use better technology.

…continued on next page …

Handling bad or missing input	░	*What is the effect of bad, incomplete, or missing source information on performance?* If the biometric is presented, the information provided is not typically bad or missing, even though the provided information may not necessarily be properly captured. Care must be taken to test the accuracy and consistency of the biometric system on the specific Client population. Procedures should address how misses and mismatches are handled (see discussion above on inflated and deflated accounting).

■	Most severe/difficult problem
▓	Moderate problem
▓	A problem
░	May be a problem
□	No problem likely, or not applicable

Figure 36. Gross Warranty assessment of using biometrics in UID technology.

BIOMETRICS (fingerprints) –COMPLIANCE (PRIVACY) STATEMENT

Intimate Stalker		*What vulnerabilities exist for the intimate stalker?* In cases where the biometric capture program can be made to work with artificial or previously captured images, rather than live capture, a problem may exist. For example, a stalker having access to a fingerprint image of a Client and the fingerprint capture program could generate a UID. The risk of such an occurrence is increasing as the number of fingerprint capture devices become more commonly used in daily life. Ways that non-live prints may be used with the biometric system should be understood and addressed.
Re-identification: Linking		*What vulnerabilities exist for re-identification of UIDs (and Dataset) using data linkage on UIDs?* As the use of biometrics becomes increasingly popular in society, the ability to link other data to biometric data increases. For example, as more people are fingerprinted and inexpensive fingerprint capture devices become increasingly common, many more databases to which to link fingerprints will exist. A UID that uses a fingerprint as source information may not necessarily store an image of the fingerprint sufficient for linking to other fingerprint databases; this depends on the specifics of the method used for constructing the UID from the fingerprint. Care should be taken to understand this method and related risks. The fingerprint databases maintained by law-enforcement require particular consideration. For example, one cannot simply refuse to obey a court order demanding copies of captured Client fingerprints, the UID construction method, and all associated UIDs for the purpose of matching Client prints against a criminal database. On the other hand, if the database did not exist, no such request could be made. A privacy policy and notice informing Clients of potential risks should be considered.
Re-identification: Dictionary Attack		*What vulnerabilities exist for re-identification of UIDs (and Dataset) using a dictionary attack on UIDs?* In the general case, exhaustive search is not likely though this should be confirmed in any particular solution proposed. However, a dictionary attack using a large biometric population database (e.g., law-enforcement fingerprint database) may re-identify Clients whose fingerprints are already captured there. Risks associated with linking prints with law-enforcement data should be assessed, and consideration given to the possibility of receiving a court order for such. In these cases, the method that related prints to UIDs would be used with image not live-scan data, a difference which may matter to some proposed solutions. A privacy policy and notice informing Clients of potential risks should be considered.
Re-identification: Reversal		*What is involved in reverse engineering the UID construction method?* Reverse-engineering a method that converts a biometric to a UID is not necessarily as fruitful as just using the method to make the associations (see linking and dictionary attack above). However, if the UID method requires live scan capture, motivation exists to perform the reversal.

…continued on next page …

Exposure	■	*What legal or technical risks or liabilities may be introduced based on the existence of the resulting database or UID technology?* The existence of captured biometrics on Clients can expose Client information to be the subject of court orders and search by law-enforcement and others.

■	Most severe/difficult problem
▓	Moderate problem
▓	A problem
░	May be a problem
□	No problem likely, or not applicable

System Trust
Which parties are heavily trusted?

Shelters and Planning Offices are heavily trusted to design systems in such a way that either linkages to law-enforcement databases are highly unlikely, or the Client is clearly informed.

Figure 37. Gross Compliance assessment of using biometrics in UID technology.

6.6 Consent (permission technology)

"Consent" as a UID method refers to a permission technology. The database technology that stores Client information at Shelters includes a permission flag which records whether a Client has granted permission to have her data forwarded to a Planning Office. Only the information of Clients who have granted permission is forwarded. The information of all other Clients is not forwarded. Figure 38 provides an example in which Ann and Claire have granted permission, and therefore their information is forwarded, but Betty and Donna have not granted permission, so their information is not forwarded.

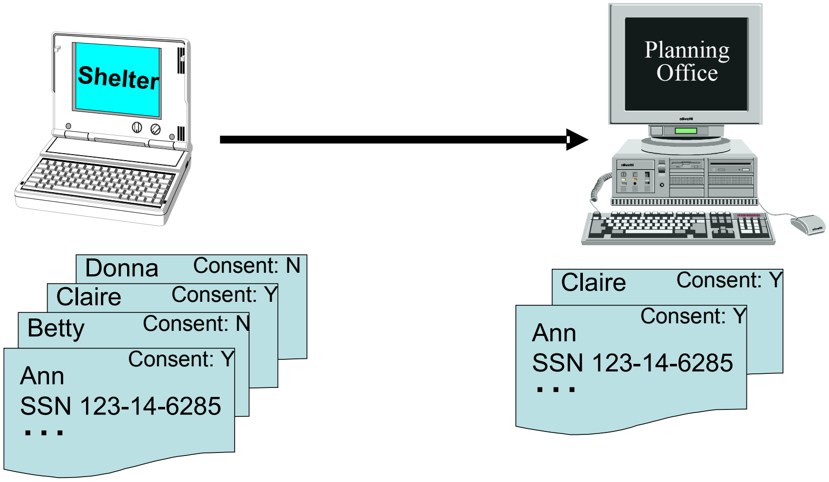


Figure 38. Consent used as the basis for deciding which Client information is forwarded to the Planning Office. Information provided to the Planning Office is explicitly identified by name and Social Security number.

Information provided to the Planning Office when consent is used typically has explicitly identified UIDs, such as name and Social Security numbers. Of course, some other UID could be used, but such cases are covered in those sections of this writing. This section addresses the situation in which the basis of de-duplication is matching explicitly identified information (e.g., name and Social Security number) that is made available because the Client has granted permission for its use.

De-duplication involves matching explicitly identified information, such as names; but matching names is horribly problematical. Clients may use nicknames or exchange first and middle names. Misspellings may be common. A well-known de-duplication method used for matching names is Soundex, which matches spellings that may look or sound similar [24]. Using Soundex, the names "James" and "John" are hashed to J52 and J5, respectively, but the names "John," "Jane" and "Jean" are all hashed to the same "J5" value. Therefore, Soundex can frustrate de-duplication.

Of course, consent allows more identifying fields to be shared, so de-duplication problems experienced with name-only matching, for example, may be augmented to exploit multiple fields of information in an attempt to account for recording errors. It should be noted however, that methods that perform such matching reliably are not trivial [25] and should be used with care.

Consent as a UID technology places Clients in the situation of sharing risks and liabilities with Shelters and Planning Offices. The use of explicit UIDs dramatically increases risks for Clients over that of other UID technologies, so standard privacy policy notices discussed earlier in Section 4 are not sufficient; more rigorous versions are needed. It is important to completely and accurately disclose the uses of Dataset and circumstances of sharing. Clients should understand HMIS data uses as well as any secondary data uses of Dataset. (Secondary uses are those situations in which Dataset, in part or whole, is shared beyond the HMIS context.) Clients must be sufficiently informed beforehand of data sharing practices; and conversely, Shelter and Planning Office practices must respect and enforce this originally agreed upon characterization.

Handling situations in which Clients do not grant permission must be considered. Clients cannot be coerced into providing permission, and Clients cannot be denied services for refusing to grant permission. Yet, Clients who do not grant permission deflate the accounting.

Inconsistent permissions may go undetected. A Client may grant permission at one Shelter and not at another, thereby providing an incomplete accounting. These situations should be considered, as well as the ability of a Client to revoke permission previously granted and vice versa.

See Figure 39 and Figure 40 for a gross assessment of using consent as a UID technology. Issues related to utility and the warranty statement appear in Figure 39. Issues related to privacy and the compliance statement appear in Figure 40. While shadings may identify some problems as being of severe or moderate concern, these problems may be sufficiently addressed with straightforward practices, policies, or technology decisions.

(Additionally, consent is not allowed under VAWA; see Section 7.2.1.)

CONSENT –WARRANTY (UTILITY) STATEMENT

Non-Verifiable source information	■	*If a UID is based on non-verifiable source information provided by the Client that is not truthful or is inconsistently used, what happens?* Serious de-duplication problems are likely if Clients provide non-verifiable source information inconsistently. On the other hand, source information that is not truthful, but consistently provided, is typically not a problem.
Verifiable source information	□	*Can problems occur if the UID is based on verifiable source information?* Using invariant Client information that can be consistently verified on each visit is likely to avoid problems. Even if the information is not correct, but consistently verified on each visit, no problems are likely. An example of invariant verifiable Client information can be a Social Security number verified to a Social Security card, or a driver's license number. A biometric could also be used.
Client confidence and trustworthiness	░	*How trustworthy is the UID likely to be perceived by Clients (as well as by those who regularly intake Clients)?* Requesting the Client's consent to share captured information tends to build Client confidence because Clients tend to feel in control of their information and believe that the process is transparent. In reality, the consent may place no limits on secondary sharing beyond the HMIS context and intake personnel may learn such. Care should be taken that the accompanying consent form and privacy notices accurately inform Clients of actual data flow, sharing practices, privacy safeguards, and Client options.
Inflated accounting	▓	*What are the circumstances under which de-duplication is likely to inflate the accounting?* Because of the increased Client confidence consent may elicit, Clients may be more willing to provide more sensitive detailed information than with other technologies, but having more information on which to match Client visits does not necessarily lead to more accurate de-duplication. The specifics of how de-duplication is performed matters. For example, name matching can be particularly problematical because of variations in the ways Clients may present their names (e.g., interchanging first and middle names, using nicknames, or different last names), not to mention typographical errors. Using an accurate de-duplication instrument is important.
Deflated accounting	▓	*What are the circumstances under which de-duplication is likely to deflate the accounting?* As was stated above with inflated accounting, having more information on which to match Client visits does not necessarily lead to more accurate de-duplication. The specifics of how de-duplication is performed matters. For example, name matching using crude algorithms like Soundex can inappropriately match names of different Clients together. Using an accurate de-duplication instrument is important. Clients who do not grant consent can deflate accounting, so additional procedures are needed to handle these cases.

…continued on next page …

Handling bad or missing input	░	*What is the effect of bad, incomplete, or missing source information on performance?* While bad or missing information is always possible, more identifying information is typically collected in these environments allowing for a larger number of data elements to be alternatively used for matching in cases where some information is bad or missing. Name matching tends to be problematical, as discussed, but having more fields on which to compare can help.

■	Most severe/difficult problem
▓	Moderate problem
▓	A problem
░	May be a problem
□	No problem likely, or not applicable

Figure 39. Gross Warranty assessment of using consent as a UID technology.

CONSENT –COMPLIANCE (PRIVACY) STATEMENT

Intimate Stalker	■	*What vulnerabilities exist for the intimate stalker?* Because consent tends to allow the collection of more sensitive information, anyone with access can be potentially compromised by the stalker to gain access. Further, secondary sharing tends to increase the number of copies of the information appearing beyond the HMIS context, which in turn, increases the number of people having access.
Re-identification: Linking	■	*What vulnerabilities exist for re-identification of UIDs (and Dataset) using data linkage on UIDs?* Because of the increased Client confidence the consent approach may elicit, Clients may be more willing to provide more sensitive detailed information than with other technologies, and the UID itself is explicitly identifying, thereby making linking a serious problem.
Re-identification: Dictionary Attack	■	*What vulnerabilities exist for re-identification of UIDs (and Dataset) using a dictionary attack on UIDs?* Because demographics and more sensitive information tends to be stored, a dictionary attack per se appears similar to linking the information to a large, population-based database, which can pose serious problems.
Re-identification: Reversal	■	*What is involved in reverse engineering the UID construction method?* The UID is an explicit identifier (e.g., Social Security number), so there is nothing to reverse. The UID itself reveals the sensitive information that would be the object of the reversal.
Exposure	■	*What legal or technical risks or liabilities may be introduced based on the existence of the resulting database or UID technology?* The existence of demographics and sensitive information on Clients can expose Client information to court orders and search by law-enforcement and others. It is more likely to draw requests for research purposes and administrative oversight in its explicitly identified form. Practices and policies for de-identification and secondary use should be considered. A privacy policy informing Clients of potential risks should be considered.

■	Most severe/difficult problem
▓	Moderate problem
▓	A problem
░	May be a problem
□	No problem likely, or not applicable

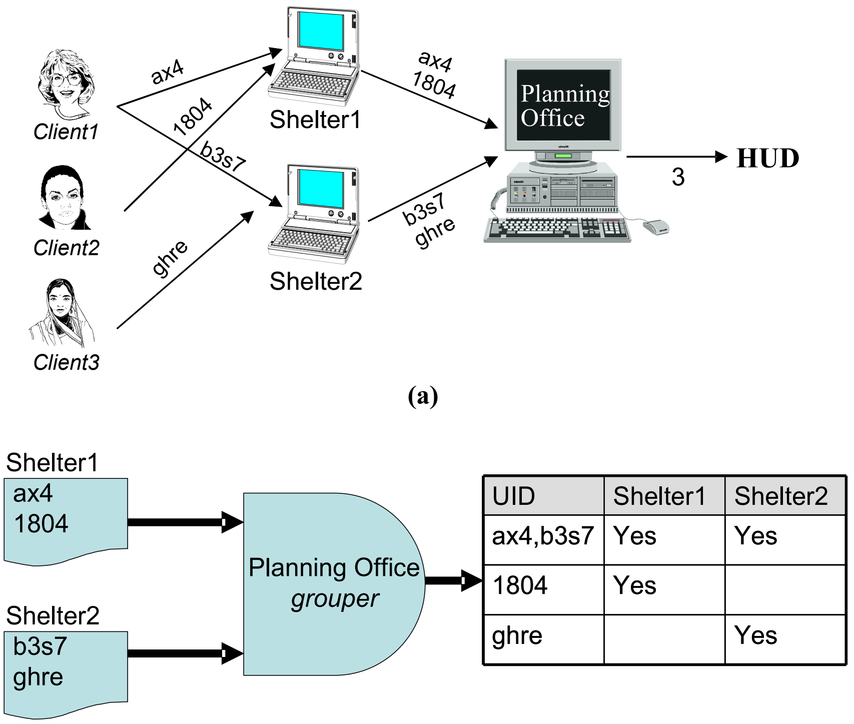
System Trust
Which parties are heavily trusted?

Planning Offices are heavily trusted with the explicitly identified Client data.

Figure 40. Gross Compliance assessment of using consent as a UID technology.

6.7 Inconsistent hashing

Inconsistent hashing works similar to regular hashing (Section 6.2) except each Client gets a different hash number at each Shelter. The Planning Office has a special methods that groups UIDs for the same Clients together ("grouper"). Figure 39 shows different Clients visiting different Shelters. Each Client is assigned a different UID at each Shelter, thereby providing an inability to link information across Shelters without the special grouping method available to the Planning Office. The Planning Office is able to use its grouping method to link UIDs belonging to the same Clients.



(a)

UID	Shelter1	Shelter2
ax4,b3s7	Yes	Yes
1804	Yes	
ghre		Yes

(b)

Figure 41. Depiction of inconsistent hashing used as a UID technology. Above (a) shows Clients assigned different UIDs at Shelters, which are forwarded to the Planning Office. Below (b) shows the Planning Office using a special method to group UIDs belonging to the same Clients.

Inconsistent hashing can be achieved in a variety of ways that primarily differ by the amount of trust given the Planning Office, which holds the grouping method [26].

The most naïve approach, which should be avoided, uses public key encryption. The Planning Office issues a public key unique to each Shelter. UIDs are encrypted with the Shelter keys, making each UID Shelter specific. Because the Planning Office has the matching private key for each Shelter, the Planning Office can reveal the original UID source information, which is then used for direct matching. This approach has the undesirable side effect that the source information (e.g., Social Security number) is revealed to the Planning Office.

A better approach uses strong hashing (Section 6.2) to protect source information from being explicitly revealed, but this approach requires more computation. Each Shelter has a unique strong hash function to generate Client UIDs. The Planning Office holds a copy of each Shelter's hash function. After the Shelters provide their UIDs, the Planning Office hashes the UIDs by every other Shelter's hash function. This takes advantage of the property that the order in which hashes of hash values are performed does not matter. For example, consider Figure 41:

Shelter 1's hash of b3s7 = Shelter 2's hash of ax4
but
Shelter 1's hash of ghre O Shelter 2's hash of ax4 or 1804.

There is concern with this approach. Because the Planning Office has a copy of each Shelter's hash function, a dictionary attack at the Planning Office is possible.

See Figure 42 and Figure 43 for a gross assessment of using consent as a UID technology. Issues related to utility and the warranty statement appear in Figure 42. Issues related to privacy and the compliance statement appear in Figure 43. While shadings may identify some problems as being of severe or moderate concern, these problems may be sufficiently addressed with straightforward practices, policies, or technology decisions.

INCONSISTENT HASHING –WARRANTY (UTILITY) STATEMENT

Non-Verifiable source information	■	*If a UID is based on non-verifiable source information provided by the Client that is not truthful or is inconsistently used, what happens?* Serious de-duplication problems are likely if Clients provide non-verifiable source information inconsistently. On the other hand, source information that is not truthful, but consistently provided, is typically not a problem.
Verifiable source information	□	*Can problems occur if the UID is based on verifiable source information?* Using invariant Client information that can be consistently verified on each visit is likely to avoid problems. Even if the information is not correct, but consistently verified on each visit, no problems are likely. An example of invariant verifiable Client information is a reliably captured biometric.
Client confidence and trustworthiness	▒	*How trustworthy is the UID likely to be perceived by Clients (as well as by those who regularly intake Clients)?* Like hashed UIDs, inconsistently hashed UIDs tend to appear cryptic, which can instill Client and intaker confidence and thereby avoid problems. Further, because UIDs are different across Shelters (and can even be different on multiple visits to the same Shelter), additional Client and intaker confidence can be attained. Problems may emerge based on the sensitivity of requested source information despite the cryptic appearance of the UID itself. Educating Clients and those who perform intake regularly and/or issuing privacy notices may help.
Inflated accounting	■	*What are the circumstances under which de-duplication is likely to inflate the accounting?* Count inflation can occur in cases where a Client provides different source information on different visits. In these cases, different UIDs are generated and therefore will not match to each other even though they are assigned to the same Client. Count inflation can also occur in cases in which a Client provides incomplete or missing information or different source information on different visits, thereby producing different UIDs across Shelters.
Deflated accounting	▒	*What are the circumstances under which de-duplication is likely to deflate the accounting?* Count deflation is possible when different Clients provide identical complete and incomplete information. A glaring example occurs for Clients in which all relevant source information is missing. Attention should be paid to how these situations are addressed in UIDs across Shelters. Count inflation is more likely than deflation.

…continued on next page …

Handling bad or missing input	■	*What is the effect of bad, incomplete, or missing source information on performance?* Typing mistakes and incomplete or missing information can generate different UIDs for a Client than would have been generated with complete and properly entered information. This tends to inflate accounting by generating spurious UIDs for Clients having multiple visits. Incomplete and missing information also tend to inflate accounting. Inflation is more likely than deflation.

■	Most severe/difficult problem
▓	Moderate problem
▓	A problem
░	May be a problem
	No problem likely, or not applicable

Figure 42. Gross Warranty assessment of using inconsistent hashing as a UID technology.

INCONSISTENT HASHING –COMPLIANCE (PRIVACY) STATEMENT

Intimate Stalker	▓	*What vulnerabilities exist for the intimate stalker?* Because each Shelter has a different UID for the same Client, access to Shelter information is limited to a Shelter-by-Shelter basis. Vulnerabilities that are able to be exploited by an intimate stalker are limited to the Planning Office, which controls the grouping method. Vulnerabilities at the Planning Office may be addressed by control and audit of the grouping method and grouped UIDs.
Re-identification: Linking	□	*What vulnerabilities exist for re-identification of UIDs (and Dataset) using data linkage on UIDs?* Because a different UID is generated at each Shelter a Client visits, and the UIDs are not used outside HMIS data, unauthorized linking is not likely. Practices should limit and account for hash function use.
Re-identification: Dictionary Attack	▓	*What vulnerabilities exist for re-identification of UIDs (and Dataset) using a dictionary attack on UIDs?* Because a different UID is generated at each Shelter a Client visits, and the UIDs are not used outside HMIS data, a dictionary attack is not likely to be fruitful except at the Planning Office. Colluding Shelters (or access to the Planning Office's grouper) can lead to re-identifications. Vulnerabilities at the Planning Office may be addressed by control and audit of the grouping method and grouped UIDs.
Re-identification: Reversal	░	*What is involved in reverse engineering the UID construction method?* When using strong hash functions, reversal is not usually an issue. But if the Shelters' hash functions are available to unlimited use by the Planning Office, care must be taken to control or limit hash function use to avoid unwanted dictionary attacks (discussed above) or reverse compilations. (A dictionary is more likely than an attempt to reverse compile the function.)
Exposure	□	*What legal or technical risks or liabilities may be introduced based on the existence of the resulting database or UID technology?* The existence of inconsistently hashed UIDs used only in the HMIS-context is not likely to expose Clients to additional risks beyond those mentioned above.

■	Most severe/difficult problem
▓	Moderate problem
▓	A problem
░	May be a problem
□	No problem likely, or not applicable

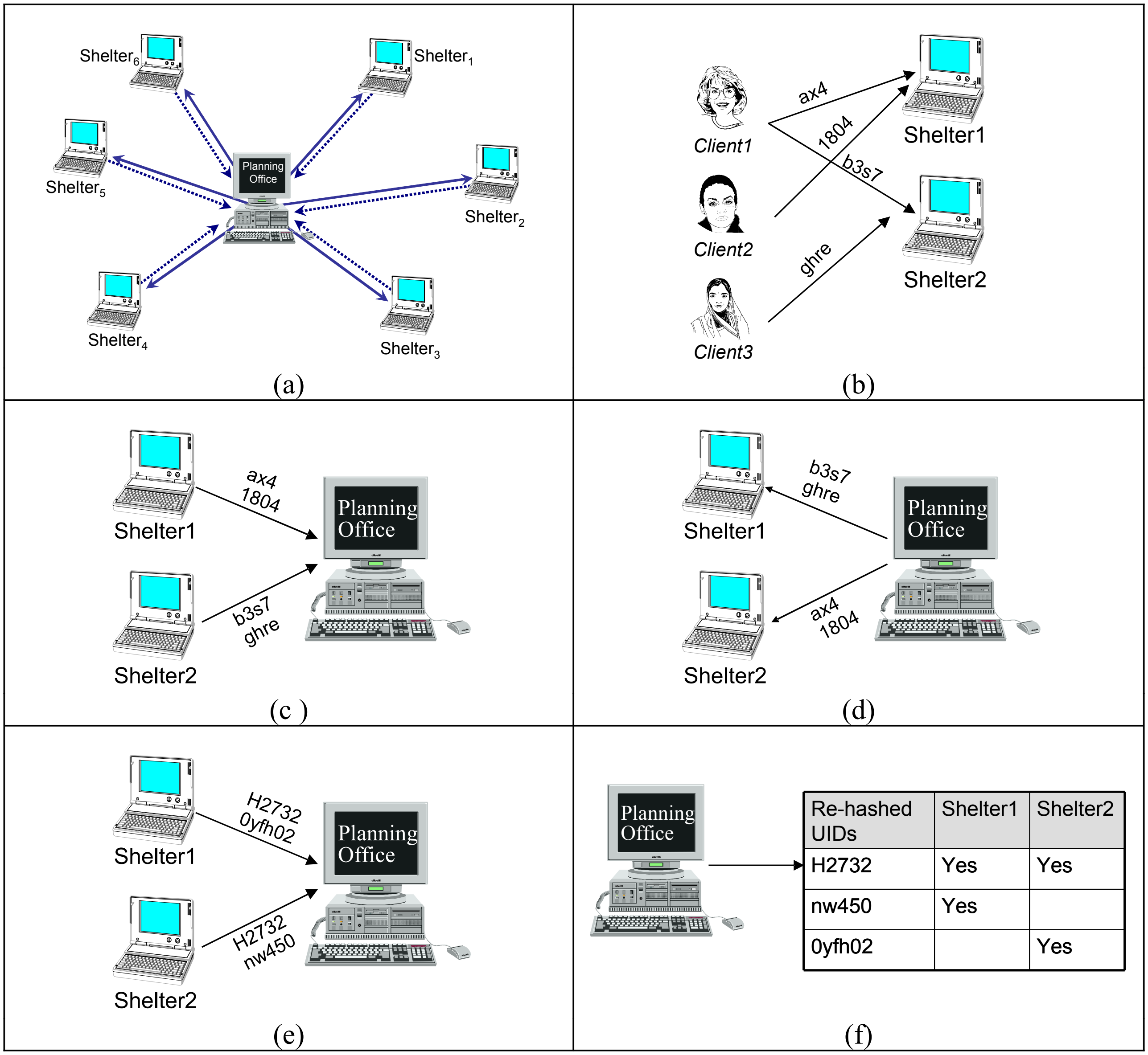
System Trust
Which parties are heavily trusted?

Planning Offices are heavily trusted to control access and use of the grouping method that links different UIDs to the same Clients.

Figure 43. Gross Compliance assessment of using inconsistent hashing as a UID technology.

6.8 Distributed query

Using distributed query, de-duplication is done on Shelter computers interacting with the Planning Office computer over a network. There are multiple ways this can be achieved. An example analogous to answering AHAR questions (Section 3.6) directly over the network is available at [27]. Another way to use distributed query is described in Figure 44 using an approach that resembles inconsistent hashing (Section 6.7) except the hash functions remain on the Shelter computers.



Re-hashed UIDs	Shelter1	Shelter2
H2732	Yes	Yes
nw450	Yes	
0yfh02		Yes

Figure 44. Distributed query (a) overview showing that Shelter computers communicate directly with the Planning Office computer. A step-by-step example of de-duplication appears in (b) through (f). Clients appear at Shelters in (b). Shelters report inconsistent hashed UIDs to Planning Office in (c). Planning Office requests each Shelter to compute the hash of every other Shelter's UIDs in (d) and Shelters respond in (e). Planning Office then compares results in (f).

In Figure 44 (b), Clients are given unique UIDs at each Shelter using strong hash functions (Section 6.2). Client 1, for example as UID ax4 at Shelter 1 and b3s7 at Shelter 2. UIDs are reported to the Planning Office in (c). The Planning Office then sends the UIDs to all the other Shelters to be re-hashed in (d). This takes advantage of the property that the order in which hashes of hash values are performed does not matter.

Shelter 1's hash of b3s7 = Shelter 2's hash of ax4
but
Shelter 1's hash of ghre O Shelter 2's hash of ax4 or 1804.

In (e), the Shelters provide the re-hashed UIDs back to the Planning Office, which matches them in (f) to show distinct visit patterns.

One concern with this system is the need to have Shelter computers on-line. One never knows when a machine may become unavailable due to repair. One strategy to limit availability problems is to perform the computation monthly, so that interim values can be used to offset any missing information needed for the yearly accounting. In locations where Shelters tend to use commercial or the same service providers to maintain Client data, Shelter information should be reliably available.

See Figure 45 and Figure 46 for a gross assessment of using consent as a UID technology. Issues related to utility and the warranty statement appear in Figure 45. Issues related to privacy and the compliance statement appear in Figure 46. While shadings may identify some problems as being of severe or moderate concern, these problems may be sufficiently addressed with straightforward practices, policies, or technology decisions.

DISTRIBUTED QUERY –WARRANTY (UTILITY) STATEMENT

Non-Verifiable source information	■	*If a UID is based on non-verifiable source information provided by the Client that is not truthful or is inconsistently used, what happens?* Serious de-duplication problems are likely if Clients provide non-verifiable source information inconsistently. On the other hand, source information that is not truthful, but consistently provided, is typically not a problem.
Verifiable source information		*Can problems occur if the UID is based on verifiable source information?* Using invariant Client information that can be consistently verified on each visit is likely to avoid problems. Even if the information is not correct, but consistently verified on each visit, no problems are likely. An example of invariant verifiable Client information is a reliably captured biometric.
Client confidence and trustworthiness		*How trustworthy is the UID likely to be perceived by Clients (as well as by those who regularly intake Clients)?* The fact that data are minimally shared from locally stored Shelter data tends to build Client and intaker confidence sufficient to avoid problems. Care should still be taken to limit the sensitivity of requested source information regardless. Educating Clients and those who perform intake regularly and/or issuing privacy notices may help.
Inflated accounting	■	*What are the circumstances under which de-duplication is likely to inflate the accounting?* Count inflation can occur in cases where a Client provides different source information on different visits. In these cases, different UIDs are generated and therefore will not match to each other even though they are assigned to the same Client. Count inflation can also occur in cases in which a Client provides incomplete or missing information or different source information on different visits, thereby producing different UIDs across Shelters.
Deflated accounting	▒	*What are the circumstances under which de-duplication is likely to deflate the accounting?* Count deflation is possible when different Clients provide identical complete and incomplete information. A glaring example occurs for Clients in which all relevant source information is missing. Attention should be paid to how these situations are addressed in UIDs across Shelters. Count inflation is more likely than deflation.

…continued on next page …

Handling bad or missing input	■	*What is the effect of bad, incomplete, or missing source information on performance?* Typing mistakes that are go uncorrected, as well as incomplete or missing information, can generate different UIDs for a Client than would have been generated with complete and properly entered information. This tends to inflate accounting by generating spurious UIDs for Clients having multiple visits. Incomplete and missing information also tend to inflate accounting. Inflation is more likely than deflation.

■	Most severe/difficult problem
▓	Moderate problem
▓	A problem
░	May be a problem
□	No problem likely, or not applicable

Figure 45. Gross Warranty assessment of using distributed query as a UID technology.

DISTRIBUTED QUERY –COMPLIANCE (PRIVACY) STATEMENT

Intimate Stalker	░	*What vulnerabilities exist for the intimate stalker?* because information is locally stored at Shelters and UIDs are only generated and used during sharing, a problem is not likely. Access to information is limited to a Shelter-by-Shelter basis.
Re-identification: Linking		*What vulnerabilities exist for re-identification of UIDs (and Dataset) using data linkage on UIDs?* Because information is kept under Shelter control, unauthorized linking beyond the Shelter itself is highly unlikely. It should be noted that Shelters have always had the ability to link Client data, irregardless of HMIS, because Shelters tend to capture complete, explicitly identified information.
Re-identification: Dictionary Attack		*What vulnerabilities exist for re-identification of UIDs (and Dataset) using a dictionary attack on UIDs?* Because information is kept under Shelter control, a dictionary attack is highly unlikely.
Re-identification: Reversal		*What is involved in reverse engineering the UID construction method?* Because strong hashing is used and information is kept under Shelter control, there is no globally available "UID" per se so there is nothing to reverse. If strong hashing is not used, then vulnerabilities may exist (see Section 6.2).
Exposure		*What legal or technical risks or liabilities may be introduced based on the existence of the resulting database or UID technology?* The existence of information locally controlled by Shelters is not likely to expose Clients to additional risks than already exists with storage and use of Shelter information.

■	Most severe/difficult problem
▩	Moderate problem
▩	A problem
░	May be a problem
□	No problem likely, or not applicable

System Trust
Which parties are heavily trusted?

Shelters are trusted to have computers on-line and available.

Figure 46. Gross Compliance assessment of using distributed query as a UID technology.

6.9 Summary Results

While many other factors determine whether a particular technology implementation is appropriate for use, the gross assessments in this section suggest that inconsistent hashing, distributed query and (regular) hashing may be easier to bundle with policies and best practices to get an effective solution. Scan cards, encryption, and biometrics create new kinds of risks to consider. Consent and encoding are technically the simplest to implement but harbor difficult dangers to overcome. Biometrics is the only technology that uses source information that does not require Clients to be trusted to provide truthful and consistent source information; all the other technologies tend to require Clients to provide non-verifiable, complete and consistent information (or confirm it) on each visit. Recall, these assessments do not consider the higher privacy standards imposed by newer regulation (VAWA). That appears in the next section. Figure 47 contains a quick summary of the results found across the gross assessment of initial UID technologies without consideration of VAWA. While shadings may identify problems as severe or moderate, these problems may be sufficiently addressed with effective practices, policies, or technology decisions.

Of course, details matter. The gross assessments could not provide a complete picture because decisions based on best practices and acceptable policies and particular technology implementations could not reasonably be included in one document. However, the gross assessments that are provided give a framework for reasoning about technical solutions and their issues in generating and matching UIDs. Lessons learned appear in Figure 48 and Figure 49.

	UTILITY						PRIVACY				
UID TECHNOLOGY	Non-verifiable source	Verifiable source	Client Trust	Inflate Accounting	Deflate Accounting	Bad or missing info	Intimate stalker	Linking	Dictionary attack	Reverse engineer	Expose new issues
Encoding	■	░	▒	░	■	■	■	■	■	■	▒
Hashing	■	□	░	░	■	■	■	■	■	▒	□
Encryption	■	□	▒	░	■	■	■	■	■	▒	▒
Scan Cards/RFID	▒	░	■	■	░	░	▒	▒	░	░	■
Biometrics	□	□	░	▒	▒	░	▒	▒	▒	▒	■
Consent	■	□	░	▒	▒	░	■	■	■	■	■
Inconsistent Hash	■	□	░	■	░	■	▒	□	▒	░	□
Distributed Query	■	□	□	■	░	■	░	□	□	□	□

■	Most severe/difficult problem
▒	Moderate problem
▒	A problem
░	May be a problem
□	No problem likely, or not applicable

Figure 47. Summary of gross assessments of UID technologies, showing utility (warranty) and privacy (compliance) issues. No consideration of the higher privacy standards imposed by VAWA appear.

Non-Verifiable source information	*If a UID is based on non-verifiable source information provided by the Client that is not truthful or is inconsistently used, what happens?* Consistent use of the UID by the Client, irregardless of whether the source information is truthful, is important for avoiding problems. As long as a Client uses the same UID and only that UID, problems can be avoided.
Verifiable source information	*Can problems occur if the UID is based on verifiable source information?* Consistency, not truthfulness, is paramount to avoiding problems. Using invariant Client information that can be consistently verified on each visit is likely to avoid problems. Even if the information is not truthful or correct, but is consistently verified on each visit, no problems are likely. Few sources of invariant verifiable source information are known; however, one such example is a reliably captured biometric.
Client confidence and trustworthiness	*How trustworthy is the UID likely to be perceived by Clients (as well as by those who regularly intake Clients)?* Instilling Client trust in the system can contribute to overall performance because Clients are more likely to provide truthful and consistent information to a system they trust. UIDs that appear to be cryptic (e.g., hashing, encryption, inconsistent hashing) can evoke more confidence than UIDs in which captured information appears transparent (e.g., encoding). Those who conduct the intake of Clients can dramatically influence the perception Clients may have of the system. Intake personnel can encourage Clients to give incorrect information, or even if Clients provide truthful information, intake personnel may record non-truthful information in a belief they are protecting Client privacy. Therefore, educating those who perform intake can be very important to overall performance.
Inflated accounting	*What are the circumstances under which de-duplication is likely to inflate the accounting?* Getting consistent source information can avoid inflated counts and conflicting Client visit information. Also, it is important to test the accuracy of the de-duplication instrument to expose problems and seek better solutions.
Deflated accounting	*What are the circumstances under which de-duplication is likely to deflate the accounting?* Getting consistent source information can avoid deflated counts and conflicting Client visit information. Also, it is important to test the accuracy of the de-duplication instrument to expose problems and seek better solutions.
Handling bad or missing input	*What is the effect of bad, incomplete, or missing source information on performance?* Unintended typing mistakes and missing information are likely to happen in real-world use. While many typing mistakes may be caught by the program in which the information is entered, some allowance has to be made for missing information. Under many real-world scenarios, it may not be possible to accurately answer the information. Therefore, consideration must be given on how to handle these cases.

Figure 48. Summary of Warranty issues found in technology assessments in Section 6.

Re-identification: Linking	*What vulnerabilities exist for re-identification of UIDs (and Dataset) using data linkage on UIDs?* Linking UIDs and Dataset to other available information requires particular attention to be paid to the demographics on which UIDs may be based. This is particularly important with hashing and encryption if access to the hash or encryption function is not controlled. For example, suppose a voter list is to be linked to a Dataset in which UIDs are hashed or encrypted using Client demographics as source information. The hash or encryption function is used with the records in the voter list to produce a UID for each record; then, the UIDs in Dataset are matched to UIDs in the voter list to re-identify Clients by name. This is a combination dictionary-attack and linking.
Re-identification: Dictionary Attack	*What vulnerabilities exist for re-identification of UIDs (and Dataset) using a dictionary attack on UIDs?* Dictionary attacks, like linking attacks, can be realized on encoded, hashing, and encryption functions, depending on the source information used and the availability of the source information in other available datasets. Controlling access to the hash or encryption function and key can help. Such control would likely be realized by forcing the function to only run on certain machines for certain named persons. All uses by those people would be logged and the logs routinely checked for inappropriate use. Other security measures can also be implemented.
Re-identification: Reversal	*What is involved in reverse engineering the UID construction method?* Reverse engineering UIDs is not typically the most fruitful kind of attack because cryptographic strong hashing and encryption methods can be used to thwart those attempts, and other approaches tend to require far less technical skill and effort. When considering these kinds of technologies, It is important to use strong methods and not homemade methods whose protection is found in the fact that they are merely unknown or obscure. A highly motivated attacker may be able to defeat these homemade attempts. Additionally, these homemade methods cannot be held to public review (as can the cryptographically strong methods) else they risk being exposed, which further limits the ability to verify the strength of their protection.
Exposure	*What legal or technical risks or liabilities may be introduced based on the existence of the resulting database or UID technology?* Some technologies generate additional kinds of risks by their existence. Scan cards can expose a Client to an intimate attacker. Encryption keys can be back doors to accessing data. The potentially increased collection of data that may be realized from consent makes the data more likely to be requested for secondary uses beyond the HMIS context; and, biometrics, especially fingerprints, can give rise to data sharing with law-enforcement, which is beyond the HMIS context.

…continued on next page …

System Trust
Which parties are heavily trusted?

Individual insiders are heavily trusted when using encoding, hashing or encryption.
System developers are trusted when strong methods are not used (hashing and encryption).
Planning Offices are heavily trusted when using consent or inconsistent hashing.
Shelter computers are heavily trusted when using distributed query.
Clients are heavily trusted when using scan cards.

Figure 49. Summary of Compliance issues found in technology assessments in Section 6.

In summary, this section provides a framework for reasoning about and assessing proposed technical solutions for generating and matching UIDs. Eight categories of technologies (encoding, hashing, encryption, scan cards/RFID, biometrics, consent, inconsistent hash, and distributed query) were examined and a set of recommendations made. While significant differences and trade-offs exist in the use of these technologies, there is no magic solution as much as best practices that must accompany any chosen technology sufficient for it to be shown that there is minimal risk of client re-identification and reasonable correctness in computing an unduplicated accounting when using the technology with accompanying practices.

7. Impact of VAWA on UID technologies

In January 2006, Congress passed The Violence Against Women and Department of Justice Reauthorization Act of 2005, H.R. 3402 ("VAWA")[10], which has a profound impact on HMIS data elements and on protecting against the previously described privacy threats. VAWA makes special provisions for a HMIS to use UIDs. Specifically, section 605 (A) states that "Victim service providers … shall… not disclose for purposes of a … [HMIS] personally identifying information about any client. …The Secretary may … require … for purposes of HMIS non-personally identifying data that has been de-identified, encrypted, or otherwise encoded.…"

VAWA defines the phrase "personally identifying information" in 605 (A) as:

> PERSONALLY IDENTIFYING INFORMATION OR PERSONAL INFORMATION.— The term 'personally identifying information' or 'personal information' means individually identifying information for or about an individual including information likely to disclose the location of a victim of domestic violence, dating violence, sexual assault, or stalking, including— ''(I) a first and last name; ''(II) a home or other physical address; ''(III) contact information (including a postal, e-mail or Internet protocol address, or telephone or facsimile number); ''(IV) a social security number; and ''(V) any other information, including date of birth, racial or ethnic background, or religious affiliation, that, in combination with any other non-personally identifying information would serve to identify any individual.

VAWA effects HMIS in two significant ways. First, VAWA supports using a UID instead of explicit identifiers. Second, VAWA requires a HMIS to use a set of data elements and a technology for processing UIDs such that no Client can be re-identified. Prior to VAWA, HUD had been following the pattern of recent U.S. privacy regulation in which technologies combine with practices and policies to provide a minimal risk of re-identification.[11] The wording of VAWA, however, insists on guaranteed protection against re-identification.

[10] This was a reauthorization of the earlier VAWA (of 1998 and then 2000). It continues to focus on ending domestic violence, sexual assault, dating violence and stalking. It sets priorities and funding levels, determines options available to victims of abuse, sets criminal justice system responses to violence, and establishes national investments in prevention. Special considerations are given to HMIS under Title VI (Housing Opportunities and Safety for Battered Women and Children), Subtitle N (Addressing the Housing Needs of Victims of Domestic Violence, Dating Violence, Sexual Assault, and Stalking), Section 605, Amendment to Section 423 of the McKinney-Vento Homeless Assistance Act (42 U.S.C. 11383).

[11] In recent U.S. privacy legislation, the notion of minimal risk of re-identification appears in the medical privacy regulation known as HIPAA (Health Information Portability and Accountability Act). Under HIPAA, 45 C.F.R. § 164.514 (b)(1)(2002), patient health data may be shared outside the patient's care if "the risk is very small that the information could be used, alone or in combination with other reasonably available information, by an anticipated recipient to identify an individual who is a subject of the information." This analysis must be based on generally accepted statistical and scientific principles, and the person who makes this finding must have "appropriate knowledge and experience applying generally accepted statistical and scientific principles and methods for rendering information not individually identifiable." Unlike HIPAA, VAWA's wording is not of minimal risk but of guaranteed protection against re-identification.

7.1 VAWA's impact on data elements

VAWA requires changes in several fields in the Dataset because some of the fields enable the Dataset to link to other available information to re-identify Clients. HUD will announce a revision to the Dataset shortly, but the following recommendations reflect relevant observations made earlier in this writing.

Recommendation #18: The fields date of birth*,* gender*, and* ZIP code of last residence *must contain less specific information than the full month, day, and year of birth, and all 5 digits of the ZIP code.*

As was shown in Figure 13, these values currently allow re-identifications, but using more general values such as *age* and the *first 3 digits of the ZIP code* significantly reduces re-identifications without affecting the utility of the information in the AHAR.

Modifications will probably exclude the PIN from the Dataset in order to limit unnecessary risk of linking the Dataset to other non-HMIS data released from the same Shelter. The *ethnicity* and *race* fields may require special handling. The Program-Specific Data Elements require additional consideration in light of other kinds of data from social service programs that a Planning Office may hold. This vulnerability may differ among municipalities and states as different kinds of secondary data from related programs are available.

Section 8 introduces PrivaMix as a UID technology hat meets the higher privacy standard established by VAWA. Section 10 gives utility and privacy results when PrivaMix was used in a real-world experiment in Iowa. Section 11 then re-examines the identifiability of HMIS data elements in light of VAWA and PrivaMix.

7.2 VAWA's impact on initial UID technologies

UID technologies that Planning Offices had previously explored for constructing, maintaining and using UIDs now face additional hurdles with the passage of VAWA. See Section 6 for an assesssment of these technologies pre-VAWA. The following subsections describe the additional difficulties faced by these technologies in attempting to comply with the privacy standard established by VAWA.

7.2.1. Consent (not under VAWA)

"Consent" as a UID technology refers to a permission technology. The database that stores Client information at Shelters includes a permission flag that records whether a Client has granted permission to have her data forwarded to a Planning Office. The Planning Office only receives the information of Clients who granted permission. If the Planning Office receives explicitly identified UIDs, such as name and Social Security numbers, from consenting Clients, VAWA does not allow this approach. (Section 6.6 describes Consent as UID technology in more detail.)

7.2.2. Biometrics (not under VAWA)

Using a biometric as source information for a UID technology has the advantage that the biometric is something always present with the Client and that typically does not change. The most common biometric is a fingerprint. It can be source information to a hash or encryption function or the fingerprint itself can be the UID. VAWA prohibits fingerprint-based UIDs if fingerprint data can match law-enforcement data. (Section 6.5 describes the use of biometrics as UID technology in more detail.)

7.2.3. Scan Cards / RFID tags (maybe okay under VAWA)

Using Scan Cards as a UID technology involves issuing a card containing a UID to each Client who presents for service. Scan cards that have a magnetic strip on one side resemble credit cards. Information is stored on the magnetic strip. Radio frequency identification (RFID) cards have no magnetic strip. Information is stored within the card. A card reader can read the information even though it is not visible to the human eye. Magnetic strips are typically readable by most card readers, not just those at the issuing Shelter. A big downside to using scan cards is not from VAWA but from practical matters such as handling lost, swapped or stolen cards.

The decision of what information appears on the card determines its acceptability under VAWA. A randomly assigned number at each Shelter is fine, but VAWA may not allow the card to include explicit identifiers and certain demographics. If Shelters share the same card and the number associated with the card appears in other data, then privacy threats may exist. (Section 6.4 describes the use of scan cards as a UID technology in more detail.)

7.2.4. Encoding (problematical under VAWA)

Using “encoding” to produce UIDs simply involves concatenating parts of source information to form a UID. De-duplication is then performed by simply matching resulting UID values. An obvious problem with encoding is that given a series of UIDs and some source information, an attacker can often deduce what parts of which source information appears in the UID. When the source information uses demographics and explicit identifiers, the encoding may be problematical under VAWA. (Section 6.1 describes encoding as a UID technology in more detail.)

7.2.5. Hashing (problematical under VAWA)

Using “hashing” to produce UIDs involves computing a number from source information. A vendor can create a hash function, but if someone is highly motivated, he can often reverse an ad hoc approach. Protection using an ad hoc hash function is good only as long as no one learns the actual hash function used. Rather than using ad hoc hash functions, cryptographically strong or one-way hash methods are highly recommended. With a strong hash function, everyone can examine the method used, but still cannot reverse the process without performing more computation than is feasible. (Section 6.2 describes hashing as a UID technology in more detail.)

If the same hash values are broadly used with Clients, then they may lead to re-identifications through linking. If the intimate stalker compromises a Shelter or the Planning Office, he can learn a targeted Client’s hashed UID and use it to locate the targeted Client. Further, if the source information is a SSN or demographics, then the Planning Office, with access to the hash function and knowledge of the kind of source information used, could re-identify all UIDs by exhaustively computing all UIDs (a dictionary attack). For these reasons, hashing, even strong hashing is problematical under VAWA.

7.2.6. Encryption (problematical under VAWA)

Using encryption as a UID method is similar to hashing except with encryption there exists a "key" such that whoever has the key can reverse the process to take a UID and reveal some (or all) of the source information that produced it. The discussion and shortcomings are the same as for hashing (above in Section 7.2.5), including VAWA concerns, with the additional consideration that only the Shelter may be able to hold the key. (Section 6.3 describes encryption as a UID technology in more detail.)

The following recommendation summarizes the findings.

Recommendation #19: *The technology used to construct and de-duplicate UIDs must satisfy VAWA's requirements limiting re-identification. Consent and biometrics appear unable to satisfy the privacy standard established by VAWA. Encoding, hashing, and encryption may enable unwanted linking, and if so, pose grave concerns in attempts to use them to satisfy VAWA's privacy standard. Scan cards and RFID tags may be used, depending on the information appearing on (or within) the card.*

Of all the UID technologies assessed in Section 6, inconsistent hashing and distributed query has the best privacy results, so it is not surprising that they form the basis for PrivaMix, which is described in the next section as a UID technology that satisfies the higher privacy standard imposed by VAWA.

8. PrivaMix, a UID Technology for VAWA

Notwithstanding the stringent re-identification standard of VAWA, this section introduces a provable privacy-preserving UID technology ("PrivaMix") for gathering service utilization patterns of domestic violence shelter clients while guaranteeing the privacy of shelter clients.

8.1 The PrivaMix approach

PrivaMix combines a form of inconsistent hashing (Section 6.7) with distributed query (Section 6.8). The same client gets different UIDs at different Shelters and can get the same UID at the same Shelter. Inconsistently assigning UIDs across Shelters in this way thwarts the linking and dictionary attack pitfalls noted earlier (in Section 4). In order for the Planning Office to then compute the unduplicated accounting of Clients across Shelters, a distributed network of Shelter machines run computations on each other's UIDs to relate which UIDs relate to the same Clients. This is done without identifying Client information to the Shelters or Planning Office. Described in this way, there are three major phases: (1) inconsistent assignment of UIDs to Clients; (2) delivery of visit information to the Planning Office; and, (3) the de-duplication of UIDs by Shelters. The next subsections further describe the approach taken in each of these phases.

8.1.1. Inconsistent assignment of UIDs

Shelters share the "PrivaMix function," which is a strong one-way function (Section 6.2) used to assign inconsistent UIDs across Shelters and reporting periods. Each Shelter customizes the PrivaMix function by using it with a privately held value the Shelter selects. This is typically a large value (perhaps 512 bits or larger) and usually selected randomly and unknown to Shelter personnel. The PrivaMix function combines the Shelter's private value with a Client's source information to generate a unique UID for the Client at the Shelter. Because different Shelters have different private values, the same Client will have different UIDs at different Shelters. Together, the Shelter's private value and the Client's source information combine to determine the Client's UID.

The Client's source information can be any appropriate information specific to the client (see Section 5). For discussion in this section, references to Client source information are to the Client's Social Security number (SSN), but using other source information is possible[12] without loss of performance or protection.

Because the PrivaMix function is strong, it is infeasible to reverse the process and learn the Client's source information (e.g., SSN) from a UID. Because each Shelter customizes its PrivaMix function by randomly selecting a secretly held large random value, the Planning Office cannot feasibly exhaust all possible combinations, thereby thwarting a dictionary attack by the Planning Office (Section 5.3). Because the UIDs are not associated with any other data, linking on UIDs is not possible.

[12] In fact, *date of birth* and *part of the first name* were used as source information in the real-world experiment reported in Section 10.

For improved privacy protection, Shelters can select different private values at each HMIS reporting session, thereby thwarting the ability to link HMIS data across reporting periods if such linking is undesired.

8.1.2. Delivery of Visit Information to the Planning Office

Once Shelters generate UIDs for Clients, they forward Client visit information along with UIDs to the Planning Office for de-duplication using a secure means (e.g., overnight delivery of a CD or over a secure Internet connection). Each record in the Dataset relates to a Client at the Shelter and includes the Client's UID. At the conclusion of this step, the Planning Office has the visit information it needs, but does not know which clients across Shelters may be the same. The Planning Office cannot de-duplicate the visits without additional processing by the Shelters.

8.1.3. De-duplication of UIDs by Shelters

In order to determine which UIDs relate to the same clients, a network of Shelter machines perform a computation on each other's UIDs. Each Shelter applies the PrivaMix function on the combination of its private value and the UIDs from the other Shelters; we term this "mixing." After all Shelters finish mixing, those results, or "complete mixes," will only be the same for those UIDs whose original Client source information was the same. These UIDs refer to the same Client.

To participate in mixing, each Shelter and the Planning Office has a computer on a reasonably secure network we term the "PrivaMix Network." The "PrivaMix Protocol" dictates communications between Shelter machines and the Planning Office machine. The purpose of the communication is have each Shelter mix (apply their customized PrivaMix function) the UIDs of all the other Shelters and keep track of which mixed results match which original UIDs. It also important that each Shelter only mix a UID once. When the protocol concludes, the UIDs across Shelters that relate to the same Client will have the same multi-mixed value, so the Planning Office can identify which records belong to the same Clients.

There are many possible functions that can serve as a PrivaMix function. While detailed requirements for a PrivaMix function appear in Section 8.3, one property is essential. A PrivaMix function must have the "commutative property" [28] in order for the Planning Office to identify which UIDs across Shelters relate to the same Client. The idea of the commutative property is that if the original source information is the same, the multi-mixed UIDs will be the same, regardless of the order in which the mixing occurs.

Before looking at two example, the following recommendations relate to using PrivaMix as a UID technology.

Recommendation #20: *When using PrivaMix as a UID technology, care should be taken to avoid multiple Shelters from having the same private value. The Shelter's private value customizes the PrivaMix function to the Shelter. If multiple Shelters inadvertently have the same private value, then those Shelters assign exactly the same UIDs to the same clients. In most uses of PrivaMix, the UIDs will only be used for one-time mixing. In these cases, it is okay if Shelters inadvertently select the same private value though the likelihood of such should be rare.*

Recommendation #21: *When using PrivaMix as a UID technology, if the visit data is transmitted to the Planning Office over the PrivaMix network of Shelter and Planning Office machines, then appropriate computer security standards for the storage of Client information should be enforced because these machines contain Client source and visit information.*

Example.

Here is an example with three Clients visiting two Shelters using integer multiplication instead of a strong one-way PrivaMix function. Because integer multiplication can be easily reversed using division, it is not "strong" and therefore cannot be used as a PrivaMix function. However, integer multiplication is commutative, so this example provides an overall demonstration of mixing.

In the first step, Shelters assign inconsistent UIDs to Clients. Figure49 provides a summary. Figure 50(a) shows three distinct Clients visiting two Shelters. A personal number appears with each Client. This is the numeric representation of the Client's source information.[13] Clients have source information 3, 7, and 11. The Client having source information 3 appears at both Shelters. The Clients having source information 7 and 11 appear at Shelter 1 and Shelter 2, respectively.

The Shelters have private values. Shelter 1's private value is 13. Shelter 2's private value is 23. In this example, integer multiplication is the function used. So, each Shelter multiplies its private value by its Client's source information to assign a UID to its Client. Figure 50(b) shows that Shelter 1 assigns the Client whose source information is 3, the UID 39 because 3 multiplied by 13 is 39. Similarly, the Client whose source information is 13 gets UID 91. Shelter 2 assigns the Client whose source information is 3, the UID 69 because 3 multiplied by 23 is 69. Similarly, the Client whose source information is 11 gets UID 253. Notice that the Client whose source information is 3 gets UID 39 at Shelter 1 and UID 253 at Shelter 2.

In the second step, Shelters forward visit information to the Planning Office. This information has a record for each Client and the Client is denoted by the assigned UID. Figure 51 depicts the flow of information from the Shelters to the Planning Office. Shelter 1 forwards two records, one for a Client with UID 39 and another for a Client with UID 91. Similarly, Shelter 2 forwards two records, one for a Client with UID 69 and another for a Client with UID 253. The Planning Office stores the UIDs from each Shelter, along with the other visit information. In Figure 51, the other visit information appears as "UDE," representing the other Universal Data Elements (see Section 3.5). At this time, the Planning Office knows there are four visits, but does not know how many Clients account for the four visits.

[13] As discussed earlier in Section 5.1.1, a Client's source information may be a Social Security number, name, or a combination of other values specific to the Client. Whether the source information is a number or text, the computer represents the information as a number. Therefore, in this section, it is proper to think of the Client's source information as a numeric value, even though its print notation may include letters and symbols.

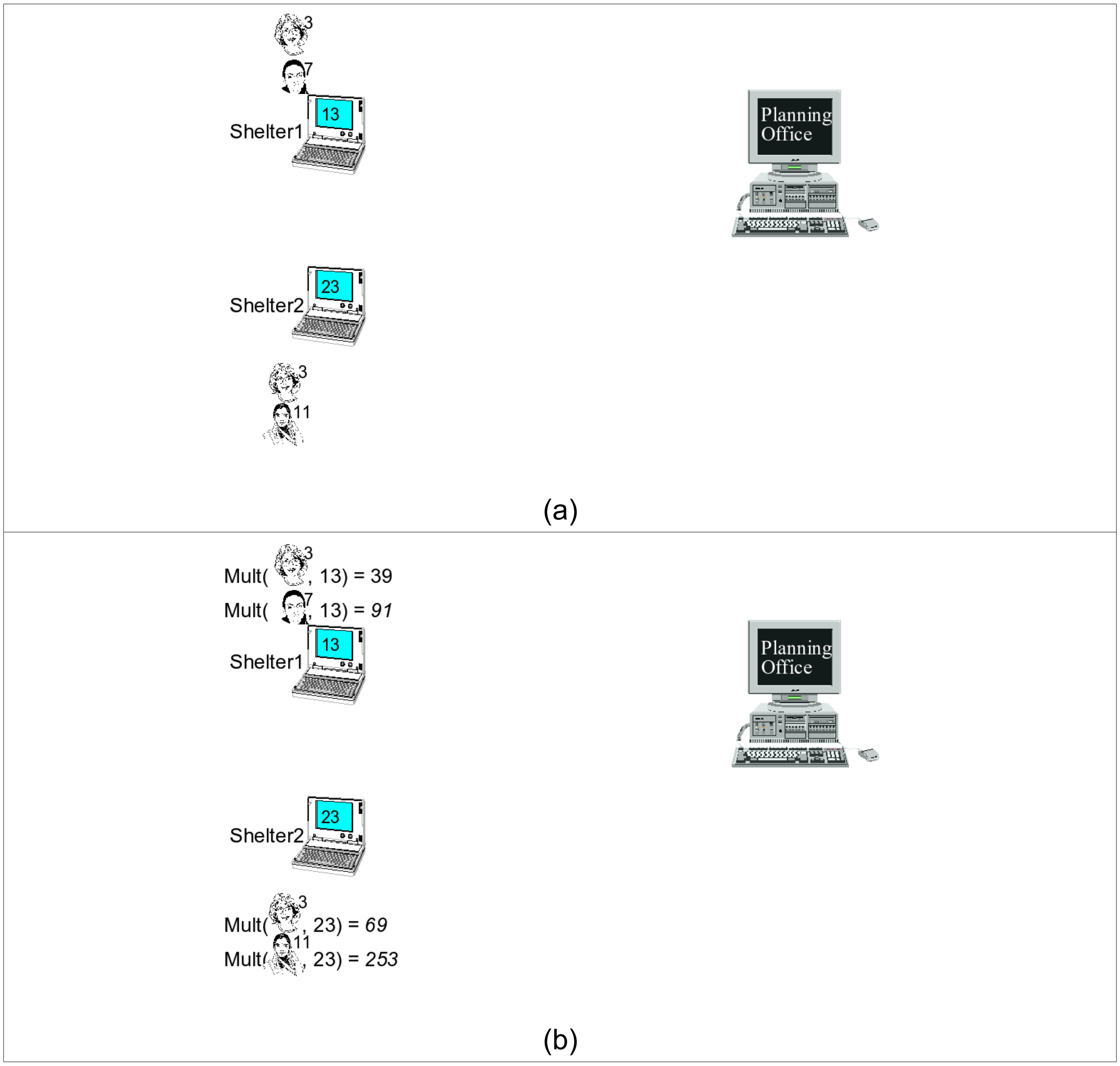


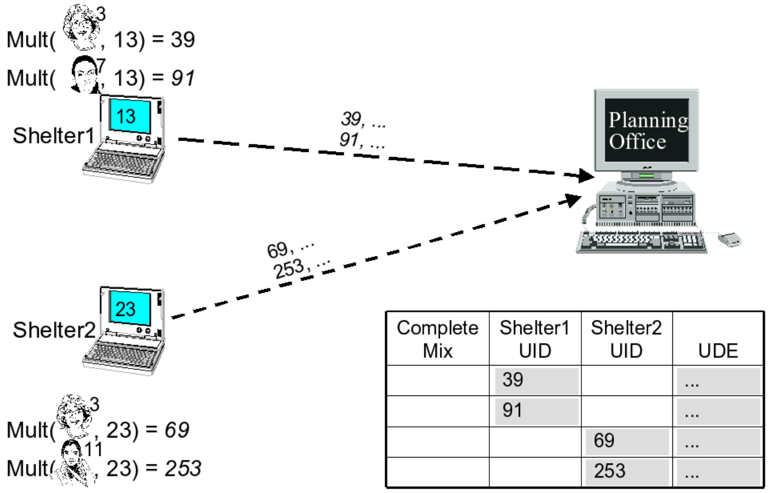
Figure 50. Shelters assign inconsistent UIDs to Clients. Shelters use integer multiplication (for example purposes only) to assign UIDs. In (a), Clients having source information 3 and 7 visit Shelter 1 and Clients having source information 3 and 11 visit Shelter 2. Shelter 1 has a private value of 13 and Shelter 2 has a private value of 23. In (b), Shelter 1 assigns UID 29 to the Client whose source information is 3 and 91 to the Client whose source information is 7. Shelter 2 assigns UID 69 to the Client whose source information is 3 and 253 to the Client whose source information is 11.

In the third step, Shelters de-duplicate the UIDs so the Planning Office can learn which visits relate to the same Clients. Figure 52 provides a step-by-step depiction. The Planning Office received UIDs 69 and 253 from Shelter 2. It forwards these to Shelter 1 for mixing; see Figure 52(a). Shelter 1 multiplies each UID by its private value 13. As shown in Figure 52(b), Shelter 1 returns the values 897, which is 13 multiplied by 69, and 3289, which is 13 multiplied by 253. The Planning Office stores the results received from Shelter 1. Because there are no other Shelters, these values are the complete mixes for the UIDs 69 and 253 received from Shelter 2.

The Planning Office received UIDs 39 and 91 from Shelter 1. It forwards these to Shelter 2 for mixing; see Figure 52(c). Shelter 2 multiplies each UID by its private value 23. As shown in Figure 52(d), Shelter 2 returns the values 897, which is 23 multiplied by 39, and 2093, which is 23 multiplied by 91. The Planning Office stores the results received from Shelter 2. Because there are no other Shelters, these values are the complete mixes for the UIDs 39 and 91 received from Shelter 1.

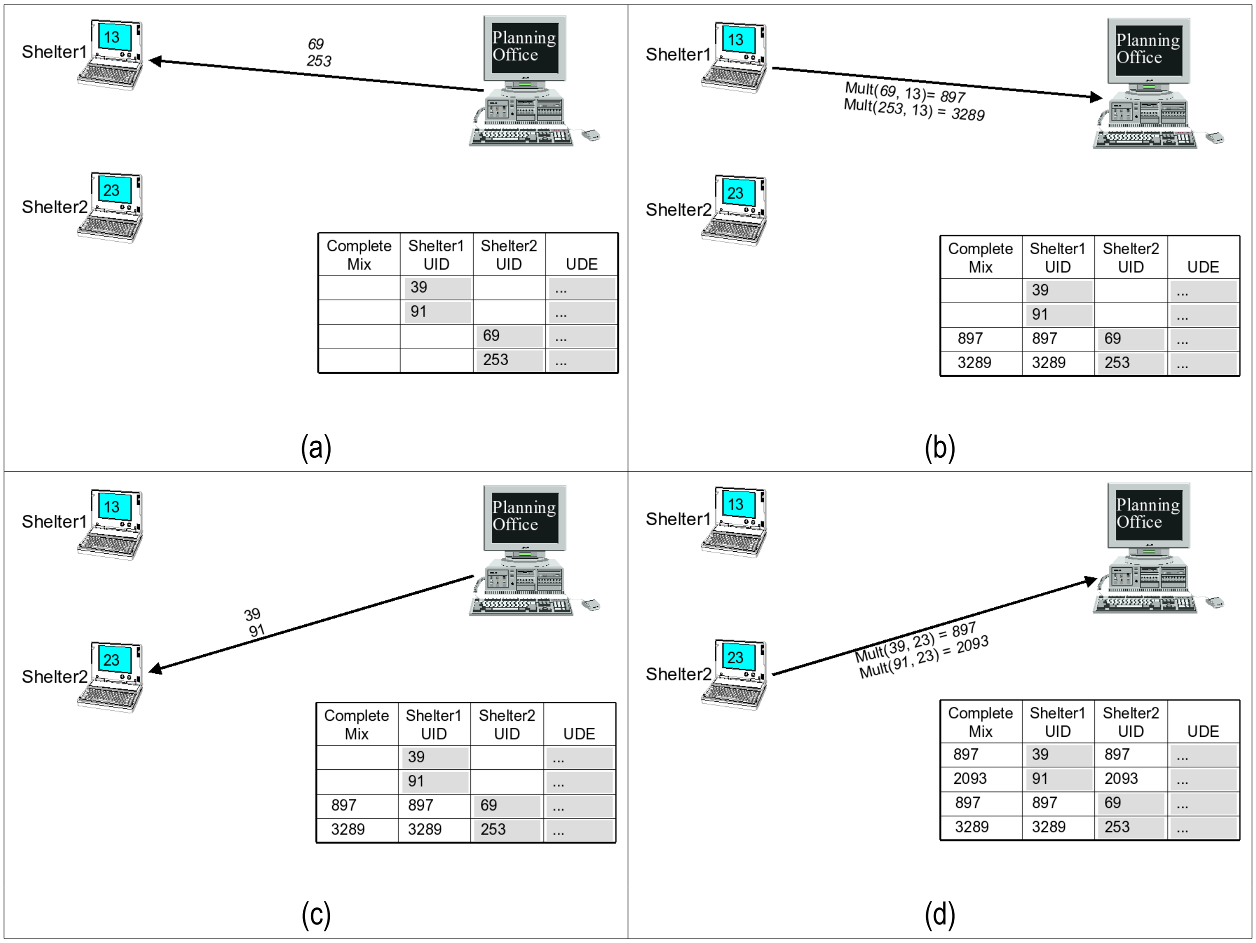
The Planning Office now learns which visits relate to the same Clients by examining which complete mixes are the same. As shown in Figure 53, two records have the same complete mix 897. One is the record with UID 39 from Shelter 1. The other is the record with UID 69 from Shelter 2. These two records relate to the same Client.

This example successfully de-duplicated the UIDs because of the commutative property of integer multiplication (used for exemplary purposes only). The order in which the multiplication occurs does not matter. When the Client whose source information is 3 visited Shelter 1 and the resulting UID 39 was mixed by Shelter 2, the result was: (3 * 13) * 23 = 897. When the same Client visited Shelter 2 and resulting UID 69 was mixed by Shelter 1, the result was: (3 * 23) * 13 = 897. Multiplying the source information by 13 and then 23 yields the same result as multiplying the source information by 23 and then 13.



Complete Mix	Shelter1 UID	Shelter2 UID	UDE
	39		...
	91		...
		69	...
		253	...

Figure 51. Shelters forward Universal Data Elements with UIDs to the Planning Office. Each record represents a Shelter visit. There are four visits, one for each of the UIDs 39, 91, 69, and 253.



(a)

Complete Mix	Shelter1 UID	Shelter2 UID	UDE
	39		...
	91		...
		69	...
		253	...

(b)

Complete Mix	Shelter1 UID	Shelter2 UID	UDE
	39		...
	91		...
897	897	69	...
3289	3289	253	...

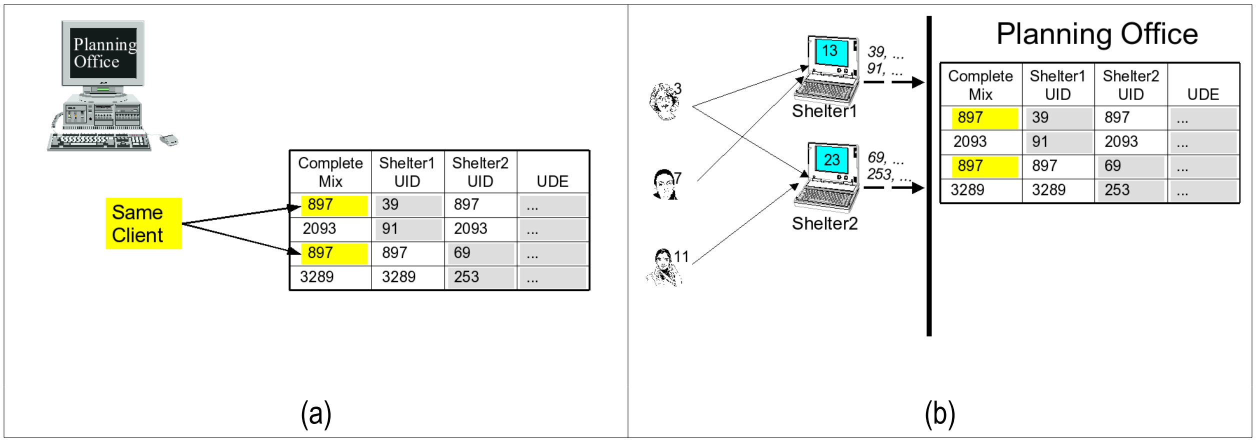
(c)

Complete Mix	Shelter1 UID	Shelter2 UID	UDE
	39		...
	91		...
897	897	69	...
3289	3289	253	...

(d)

Complete Mix	Shelter1 UID	Shelter2 UID	UDE
897	39	897	...
2093	91	2093	...
897	897	69	...
3289	3289	253	...

Figure 52. Mixing to de-duplicate UIDs. In (a), the Planning Office forwards the UIDs received from Shelter 2 to Shelter 1. In (b), Shelter 1 returns the mixed results to the Planning Office. In (c), the Planning Office forwards the UIDs received from Shelter 1 to Shelter 2. In (d), Shelter 2 returns the mixed results to the Planning Office.



(a)

Complete Mix	Shelter1 UID	Shelter2 UID	UDE
897	39	897	...
2093	91	2093	...
897	897	69	...
3289	3289	253	...

(b)

Complete Mix	Shelter1 UID	Shelter2 UID	UDE
897	39	897	...
2093	91	2093	...
897	897	69	...
3289	3289	253	...

Figure 53. Planning Office learns which records relate to the same Client. Two records have the same complete mix, 897, revealing that two of the records relate to the same Client.

Example.

Here is another example with three Clients visiting two Shelters. The example uses symbolic notation to denote the use of the PrivaMix function. Figure 54 shows Client 1 visiting both Shelters, where Client 2 only visits Shelter 1 and Client 3 only visits Shelter 2. From Client Social Security numbers (SSN), Shelter 1 produces ax4 as Client 1's UID and 1804 as Client 2's UID. Shelter 2 produces b3s7 as Client 1's UID and ghre as Client 3's UID. The generation of the UIDs depicted in Figure 54 concludes step 1, the assignment of inconsistent UIDs.

Each Shelter provides the Planning Office with the Client UID and associated Universal Data Elements for each Client that visited the Shelter. Figure 55 shows the compiled results at the Planning Office. Shelter 1 had two Clients whose UIDs are ax4 and 1804. Shelter 2 had two Clients whose UIDs are b3s7 and ghre. At this time, the Planning Office knows information about four Client visits, but does not know the total number of distinct Clients or which Clients at Shelter 1 also visited Shelter 2. This concludes Step 2, delivery of visit information to the Planning Office.

Step 3 involves de-duplicating the UIDs; see Figure 56. The Planning Office sends the UIDs received from Shelter 2, b3s7 and ghre, to Shelter 1 for mixing. Similarly, it sends the UIDs from Shelter 1, ax4 and 1804, to Shelter 2 for mixing. These appear in Figure 56(a). The result of Shelter 1's mixing of b3s7 is H2732 and of ghre is 0yfh02, as shown in Figure 56(b). Similarly, the result of Shelter 2's mixing ax4 is H2732 and of 1804 is nw450, as shown in Figure 56(b). Therefore, the results from mixing are H2732 and 0yfh02 from Shelter 1 and H2732 and nw450 from Shelter 2. Finally, the Planning Office associates the mixed values to the original UIDs to learn that ax4 and b3s7 relate to the same Client, but UIDs 1804 and ghre relate to different and distinct Clients; see Figure 56(c). The Client whose SSN was the same has the same complete mix, notwithstanding the order of mixing.

Both examples provided so far have involved only two Shelters. A larger example appears at the end of Section 8.2. It better demonstrates the scalability of mixing.

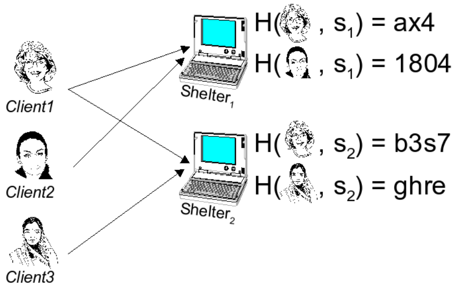
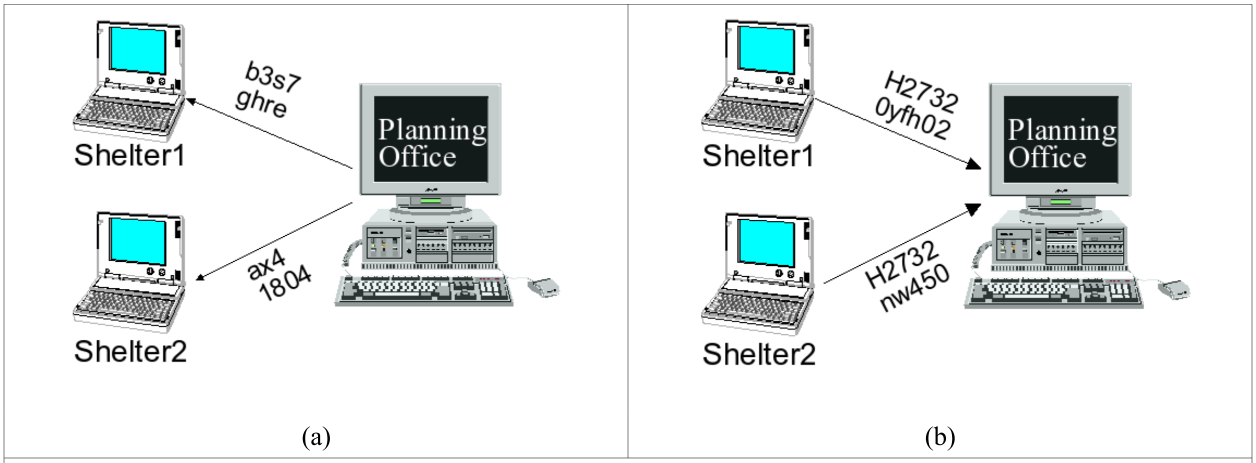


Figure 54. Each Shelter generates a UID for each Client that visits the Shelter using the Client's SSN and the a private value v_i held at the Shelter. The UID is generated using a strong, commutative PrivaMix function F. Each shelter customizes the use of the PrivaMix function because each Shelter has a different private value v_i.

Shelter	UID	UDE
Shelter 1	ax4	…
Shelter 1	1804	…
Shelter 2	b3s7	…
Shelter 2	ghre	…

Figure 55. The Planning Office compiles a table of information provided by the Shelters of Client visits. UDE refers to the Universal Data Elements that comprise the Dataset. UIDs are the Client UIDs issued at each Shelter.



(a)

(b)

Complete Mix	Shelter1 UID	Shelter2 UID
H2732	ax4	b3s7
nw450	1804	
0yfh02		ghre

(c)

Figure 56. PrivaMix Protocol for de-duplication of UIDs: (a) the Planning Office forwards UIDs to be mixed; (b) Shelters send back the mixed results; and (c) Planning Office compares complete mixes to original UIDs to learn that one Client visited both Shelters and two Clients visited one Shelter each.

8.2 Technical presentation

This subsection provides a formal description of the PrivaMix approach. Non-technical readers may advance to the claims subsection which follows this subsection without loss of understanding.

Strong Function.
A one-way (or strong) function has the property that if $\mathbf{F}$ is a strong function, $\mathbf{F}(x)=y$ computes in polynomial time, and it is computationally infeasible to compute its inverse $\mathbf{F}^{-1}(y)=x$.

The "Commutative Property".
Each Shelter j hashes a Client's source information (SSN_i) using the Shelter's private value s_j and a strong function $\mathbf{F}$ such that: $\mathbf{F}(SSN_i, s_j) = UID_{ij}$. Benaloh and de Mare [28] showed that the Shelters' private values can be chosen appropriately so that

$\mathbf{F}(\mathbf{F}(SSN_a, s_j), s_k) = \mathbf{F}(\mathbf{F}(SSN_b, s_k), s_j)$, only if $SSN_a = SSN_b$.

8.2.1. Formal Description

<u>Definitions</u>
Let $S = \{S_1, S_2, ..., S_n\}$ be n Shelters having private values $s_1, s_2, ..., s_n$, respectively.
Let $C = \{C_1, C_2, ..., C_m\}$ be m Clients having source information SSN_1, SSN_2, ..., SSN_m, respectively.
Let P be the Planning Office .
Let $\mathbf{F}$ be a strong function with the commutative property that generates the UID.
We write: $\mathbf{F}(SSN_i, s_j) = UID_{ij}$

<u>Problem Statement</u>
For each SSN_i, P learns (S_j, UID_{ij}) without re-identifying Client i or learning SSN_i.

<u>3-Step PrivaMix Protocol (also known as "PrivaMix")</u>
Step 1. Shelters compute UID_s.
For each C_i visiting Shelter S_j, Shelter S_j computes $\mathbf{F}(SSN_i, s_j) = UID_{ij}$

Step 2. Data to Planning Office.
P receives a table having attributes $\{S_j, \mathbf{F}(SSN_i, s_j), UDE_{ij}\}$ where UDE are values of the Universal Data Elements for Client C_i at Shelter S_j.
P computes multi-set[14] $\boldsymbol{H} = \{ \mathbf{F}(SSN_i, s_j) : \mathbf{F}(SSN_i, s_j) = UID_{ij}\}$

Step 3. De-duplicate UIDs.

```
 **For** $k = 1$ **to** $n$ **do**:
    $P$ computes multi-sets:
        $\boldsymbol{H}_1\{x : x = \mathbf{F}^{a}(\mathbf{F}(SSN_i, s_j)...) \text{ where } x \in \boldsymbol{H}, a \geq 0, j \neq k\}$
        $\boldsymbol{H}_2\{x : x = \mathbf{F}^{a}(\mathbf{F}(SSN_i, s_j)...) \text{ where } x \in \boldsymbol{H}, a \geq 0, j = k\}$
    $P$ sends $\boldsymbol{H}_1$ to $S_k$
    $S_k$ sends $P$:
        $\boldsymbol{H}_{Sk}\{\mathbf{F}(x, s_k) : x = \mathbf{F}^{a}(\mathbf{F}(SSN_i, s_j)...) \text{ where } x \in \boldsymbol{H}_1\}$
```

[14] A multi-set is a set in which an element may appear multiple times. $\boldsymbol{H}$, $\boldsymbol{H}_1$, $\boldsymbol{H}_2$, and $\boldsymbol{H}_{Sk}$ are multi-sets.

P computes $\boldsymbol{H} = \boldsymbol{H}_{\boldsymbol{Sk}} \cup \boldsymbol{H}_{\boldsymbol{2}}$

Example.

Here is an example involving three Clients and three Shelters. Figure 57 shows Client 1 visiting Shelters 1 and 2, where Client 2 only visits Shelter 2 and Client 3 only visits Shelter 3. Each Shelter has a strong function (**F**), having the commutative property. **F** is customized to each Shelter *j* by the Shelter' private value s_j. Each Shelter uses its customized function to compute a UID for each Client *i* using Client *i*'s source information (SSN_i). The value $\mathbf{F}(SSN_i, s_j)$ is UID_{ij} for Client *i* at Shelter *j*.

Each Shelter provides the Planning Office with the UID and associated Universal Data Elements of each Client that visited the Shelter. Figure 58 shows the compiled results in a table produced by the Planning Office.

To de-duplicate UIDs, the Planning Office sends a set of values to each Shelter to mix (apply its private value using function **F**). In Figure 59, the Planning Office sends the original UIDs from Shelter 2 and Shelter 3 to Shelter 1 for mixing. Shelter 1 sends the mixed results back to the Planning Office ending the first round of de-duplication.

In the next round of de-duplication, the Planning Office sends values to Shelter 2 for mixing; see Figure 60. The Planning Office sends the original UID from Shelter 1 for Client 1. It also sends the mixed value from Shelter 1 of the UID originally provided by Shelter 3. Together, these are values from other Shelters not yet mixed by Shelter 1.

In the final round of de-duplication, the Planning Office sends values to Shelter 3 for mixing; see Figure 61. These values are those UIDs that originated from Shelters 1 and 2 but in their current mixed form.

The rounds of de-duplication continue for as many Shelters involved. There is one round per Shelter. Each Shelter receives the UIDs originally contributed from the other Shelters, but the value used is either the original or the mixed value. When the rounds are completed, each Shelter has applied the function with its private value once on each UID, though the order in which the mixing occurred varied. The commutative property of the function guarantees that the final mixed values will be the same only if the original source information was the same. Figure 62 summarizes the findings from this example.

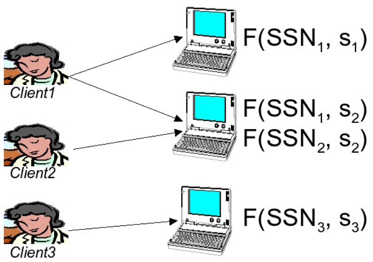


Figure 57. Each Shelter computes a UID for each Client using a strong function F, which has the commutative property. F is customized to each Shelter *j* using v_j. Client *i*'s source information is denoted as SSN_i.

Shelter	UID	UDE
Shelter 1	F(SSN1, s1)	…
Shelter 2	F(SSN1, s2)	…
Shelter 2	F(SSN2, s2)	…
Shelter 2	F(SSN3, s3)	…

Figure 58. Planning Office knowledge after receiving UIDs and Universal Data Elements from Shelters.

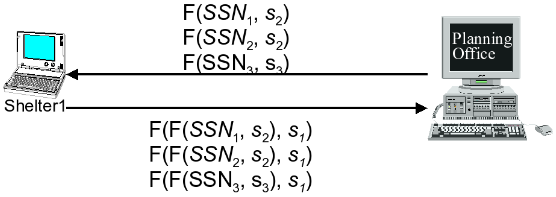


Figure 59. Round 1 of de-duplication. Original UIDs from Shelters 2 and 3 are sent to Shelter 1 for mixing.

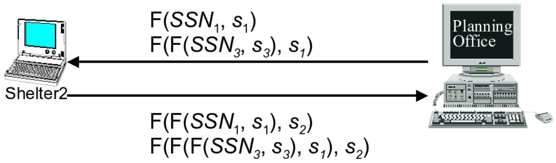


Figure 60. Round 2 of de-duplication. Shelter 2 receives the original UID from Shelter 1 and the mixed UID originating from Shelter 3.

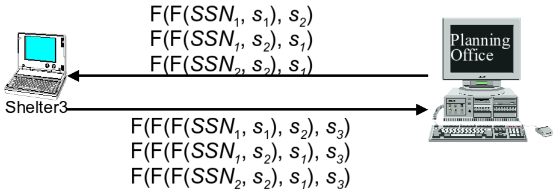

$F(F(SSN_1, s_1), s_2)$
$F(F(SSN_1, s_2), s_1)$
$F(F(SSN_2, s_2), s_1)$
Planning Office
Shelter3
$F(F(F(SSN_1, s_1), s_2), s_3)$
$F(F(F(SSN_1, s_2), s_1), s_3)$
$F(F(F(SSN_2, s_2), s_1), s_3)$

Figure 61. Round 3 of de-duplication. Shelter 3 receives the current mixed values of the UIDs that were originally contributed by Shelter 1 and Shelter 2.

$$F(F(F(\mathit{SSN}_1, s_1), s_2), s_3)$$

$$F(F(F(\mathrm{SSN}_1, s_2), s_1), s_3)$$

$$F(F(F(\mathrm{SSN}_2, s_2), s_1), s_3)$$

$$F(F(F(\mathrm{SSN}_3, s_3), s_1), s_2)$$

Figure 62. The final results from de-duplication. Values that are the same have the same source information and therefore are considered to relate to the same Client.

Below are variations to the generic PrivaMix approach. Each is described in detail following the list.

8.2.2. PrivaMix Variation 1: Shelters mix among themselves, without the Planning Office.
8.2.3. PrivaMix Variation 2: Shelters check that UIDs are legitimate
8.2.4. PrivaMix Variation 3: Matching UIDs to Universal Data Elements
8.2.5. PrivaMix Variation 4: Providing aggregate count distributions, not Client-level data
8.2.6. PrivaMix Variation 5: Anonymizing client-level data
8.2.7. PrivaMix Variation 6: Using web browsers for mixing

8.2.2. PrivaMix Variation 1: Shelters mix among themselves, without the Planning Office.

As stated above, Step 3 describes communication controlled by the Planning Office, but other models are just as valid. In the version above and in previous examples (Section 8.1), the Planning Office communicates with each Shelter in turn. Alternatively, in Step 3 the de-duplication could be done among the Shelters themselves with complete mixes and original UIDs sent to the Planning Office. Details of this variation appear below. In this writing, de-duplication assumes communication the Planning Office controls communication unless stated otherwise.

Variation of Step 3. De-duplicate UIDs.

1. Shelters randomly select a permutation of themselves. See protocol described in [29].
2. Shelters pass all (*Shelter$_i$*, *UID$_i$*, *mix$_i$*) triples to the first Shelter in the permutation for mixing. At this point, each *mix$_i$* is the same as *UID$_i$*.
3. When the Shelter has mixed each *mix$_i$*, it replaces the *mix$_i$* value in each triple with its further mixed result. Updated triples are then passed to the next Shelter in the permutation for mixing. Processing continues until the triples have complete mixes.
4. The final Shelter in the permutation forwards the final triples to the Planning Office.

The advantage of this variation to PrivaMix is that Shelters compute complete mixes without Planning Office involvement. Side effects are: (a) all Shelters learn the number of Client visit records at each Shelter; and, (b) the last Shelter(s) know the de-duplicated results.

8.2.3. PrivaMix Variation 2: Shelters check that UIDs are legitimate

Shelters can effect a check on the number of UIDs to mix by computing the total number of UIDs to mix among themselves before Step 3 of the PrivaMix Protocol begins. Then, during the PrivaMix Protocol, each Shelter can validate whether the number of values asked to mix by the Planning Office is correct. There are many ways to do this. An adaptation of the PrivaSum Protocol [27] allows the Shelters to jointly compute the total number of UIDs without Shelters knowing the number of UIDs from any particular Shelter.

8.2.4. PrivaMix Variation 3: Matching UIDs to Universal Data Elements

The description of the PrivaMix Protocol does not explain how complete mixes are matched to the original UID and UDE information provided in Step 2. There are many equally valid ways to accomplish this. A simple way is to maintain the order in which values are passed. If the Planning Office provides Shelter x with the stream v_1, v_2, v_3, ..., as values to mix, then the stream from the Shelter back to the Planning Office should be the mixes in the following order: $\mathbf{F}(v_1, s_x)$, $\mathbf{F}(v_2, s_x)$, $\mathbf{F}(v_3, s_x)$,... This approach has the advantage that Shelter and original UID information is not part of the communications.

8.2.5. PrivaMix Variation 4: Providing aggregate count distributions, not Client-level data

Rather than PrivaMix providing Client-level data to the Planning Office, PrivaMix can alternatively provide aggregate de-duplicated count distributions. As described so far, PrivaMix provides the Planning Office with a detailed visit record for each Client; this is termed Client-level data. The Universal Data Elements (Section 3.5) describe Client-level data. In this variation of PrivaMix, the Planning Office would instead get distributions of how many Clients matched particular characteristics. An example of an aggregate count distribution is a breakdown of the number of Clients in age ranges. Providing the Planning Office with aggregate count distributions gives the Planning Office exactly the information needed for reports, such as the AHAR (Section 3.6) without revealing Client-level details that can compromise privacy by enabling data linking. There are several ways to modify PrivaMix to provide aggregate count distributions. Here is one way.

In Step 2 of the PrivaMix Protocol, Shelters send Client-level data, specifically the Universal Data Elements (Section 3.5), to the Planning Office. Step 3 then involves de-duplication of UIDs only. Alternatively, Shelters can send the same Client-level information to the Planning Office, but the data is not explicitly made accessible to Planning Office personnel. The data may be held by the PrivaMix software operating on the Planning Office machine without allowing Planning Office personnel the ability to view or access the information. These data may be encrypted and/or held in temporary memory.

When UID de-duplication completes in Step 3, the PrivaMix software operating on the Planning Office machine provides aggregate de-duplicated count distributions following a configurable script that describes which combination of values to aggregate and count. For example, Figure 6 shows the kinds of questions the AHAR answers. PrivaMix could answer these questions directly to the Planning Office without sharing Client-level data.

While this variation improves privacy by being a significant guard against unwanted data linking (Section 4.2), it also limits the use of de-duplicated results. The only result is aggregate count information. Other count information would not be available.

Another concern with this variant is error-checking. Typing mistakes and inconsistent values appearing in different records relating to the same Client becomes more difficult to spot and address when personnel responsible for using the count distributions in reports cannot easily examine the data that formed the basis of the counts. Workarounds may be possible by including cross-counts and Shelter-based distributions.

8.2.6. PrivaMix Variation 5: Anonymizing client-level data

A way to help thwart data linkage threats within PrivaMix while still providing Client-level data is to anonymize the data after de-duplication. Formal protection models identify which values can be sensitive to linking and either generalize or suppress those values from the resulting dataset so that each record ambiguously relates to a minimum number of people [30][31]. For example, if a 80 year old woman is an outlier in the data because of her age, either her age would be removed from the data or generalized to a category having more people, such as "50 plus" as appropriate given the other ages appearing in the data. A downside of this approach is that Client-level data will contain generalized or suppressed values, which can make it harder to work with statistically or detect typographical errors.

The last two PrivaMix variations (Section 8.2.5 and Section 8.2.6) address ways within PrivaMix to improve privacy and help thwart data linkage threats. An alternative lies outside PrivaMix, in choosing non-identifiable Client-level data elements. Section 11 examines these trade-offs in detail.

8.2.7. PrivaMix Variation 6: Using web browsers for mixing

The generic description of PrivaMix described above establishes the need for each participant (Shelter, HMIS, and Planning Office) to have a computer on a shared network. This can be an expensive proposition if each participant needs a machine on a dedicated network, and an inexpensive proposition if each participant can alternatively use existing computers that already have Internet access. While it is reasonable to assume that virtually no participant has an extra machine available to devote to mixing alone, it is reasonable to assume that each participant already has a machine (computer or even mobile phone) for email communication and web browsing, as these have become fundamental means of sharing information. There is nothing inherent in the PrivaMix Protocol that prohibits its execution on these devices using commonly used web browsing[15] software originally shipped with these machines.

[15] A "web browser" is a computer program that allows users to view and share information over the World Wide Web. Virtually all computers, and even some mobile phones, come with web browsers. Internet Explorer is the most common web browser on machines running the Windows operating system. Other popular web browsers are Firefox and Safari.

Implementing PrivaMix through web browsers enables a wide array of existing computers to participate in mixing without installing any special software on the machines and without any special concern to user training. The important condition is that the Shelter's private value remains private to the Shelter's web browser. This is done seamlessly as described below.

When a web browser visits a website devoted to running the PrivaMix Protocol, software for producing UIDS and mixes seamlessly downloads itself into the web browser's operational environment. PrivaMix software then remains active on the machine as long as the web browser views the PrivaMix website. Once the web browser visits another website or stops running altogether, the PrivaMix software is no longer available. All machines participating in mixing should therefore have web browsers viewing the PrivaMix website throughout the process.

Here is a walk-through the PrivaMix Protocol using web browsers. To begin, all participants visit a web address[16] of a web server running the PrivaMix software. The server may run on a machine at the Planning Office. Alternatively, a third party facilitator may provide a server, which can be used by one or more Planning Offices. No additional privacy concerns result from either a Planning Office or a third party hosting the server.

Each participant must provide a previously agreed upon password to authenticate its machine; otherwise access to the PrivaMix software is not allowed. This prohibits others from wrongfully participating in a mix.

All communications between participating machines and the server are encrypted using existing web browser software. Web browsers already include encryption software. An example of its use occurs when conducting credit card and financial transactions using a web browser. Encrypting communications thwarts eavesdropping attempts.

Software containing the PrivaMix function seamlessly downloads into the Shelter's web browser environment, allowing the Shelter machine to produce UIDs and mixes as needed. After downloading the PrivaMix function, the Shelter's machine randomly selects a private value. The Shelter's web browser holds this value privately. It is never transmitted to the server.

The user of the Shelter machine selects a file to upload. This file contains Client source information and Universal Data Elements, one row for each Client. Immediately prior to forwarding a Client's Universal Data Elements to the server, the Shelter's machine computes the Client's UID and then forwards the UID and the Universal Data Elements. The web browser performs these computations automatically.

When all Shelters complete the uploading of Client information, the server contains all Client visit information but the result does not reveal which Clients across visits are the same. The Shelters must mix UIDs to de-duplicate. The server orchestrates mixing, one Shelter at a time, as described earlier. A Shelter machine uses its privately held value and copy of the PrivaMix function to mix. The final de-duplicated results appear first at the server, which then forwards the de-duplicated result to the Planning Office.

[16] A web address is often known as a URL (Uniform Resource Locator). The information stored at the web address is termed a web page. Web pages occurring from the same server idenitfy a website. Common web addresses end in .com or .edu. Examples are: www.google.com and privacy.cs.cmu.edu.

In summary, these variations, provide the following recommendations.

Recommendation #22: If desirable, use a variation of the PrivaMix Protocol to have a party other than the Planning Office orchestrate mixing. One variation (Section 8.2.2) describes how Shelters perform mixes among themselves and then forward de-duplicated results to the Planning Office. Another variation (Section 8.2.7) describes how a third-party might orchestrate de-duplication and then forward results to the Planning Office.

Recommendation #23: Thwarting data linkage threats requires further privacy consideration, realized as variations of PrivaMix and/or dictates on data elements. Rather than PrivaMix providing Client-level data to the Planning Office, PrivaMix can alternatively provide aggregate de-duplicated count distributions (Section 8.2.5). A way to help thwart data linkage threats within PrivaMix while still providing Client-level data is to anonymize the data after de-duplication (Section 8.2.6). An alternative that lies outside of PrivaMix is to chose non-identificable Client-level data elements (Section 11).

Recommendation #24: An economical implementation of the PrivaMix Protocol involves using traditional web browsers already provided with computers (Section 8.2.7). Doing so has the advantage that no dedicated machine is needed, that no additional software has to be installed, and that no intense user training is needed.

8.3 Requirements of a PrivaMix function

Given secret shelter and client information as integers, a desired PrivaMix function (**F**) must satisfy the following six requirements.

The first requirement for a PrivaMix function is as follows. Given secret source information for the client c_i and private information for any pair of shelters s_x and s_y, it should be highly unlikely that $\mathbf{F}(ci, sx) \neq \mathbf{F}(c_i, s_y)$.

The first requirement for a PrivaMix function states that the UIDs for the same client *i* appearing at different shelters *x* and *y* should not be the same. It should be highly likely that $u_{ix} \neq u_{iy}$ where $\mathbf{F}(c_i, s_x) = u_{ix}$ and $\mathbf{F}(c_i, s_y) = u_{iy}$. This is the “inconsistent assignment” requirement.

The second requirement for a PrivaMix function is that it has the property that $\mathbf{F}(c_i, s_j)=u_{ij}$ computes in polynomial time, and it is computationally infeasible to compute its inverse $\mathbf{F}^{-1}(u_{ij})=(c_i, s_j)$.

The second requirement states that **F** must be a one-way function[17]. Applying **F** to secret client and shelter information should compute in real-time. However, given a result of **F**, the secret client and shelter information should not compute in any reasonable time. This is the “one-way” requirement.

[17] The cryptography community commonly uses the term “one-way function” to refer to a hash or encryption function for which it is reasonably fast to compute the hash or encryption value, but computationally infeasible to reverse the process.

Given a client c_i, let the term "complete mix" over n shelters refer to z_i such that:
$\mathbf{F}_n(\ldots\mathbf{F}_2(\mathbf{F}_1(c_i, s_1), s_2), \ldots, s_n) = z_i$

Let the term "sub-mix" refer to $\mathbf{F}_y(\ldots\mathbf{F}_x(c_i, s_x)\ldots, s_y) = z_{it}$ where t is the sequence of shelters involved in the mix, $|t| < n$, and no shelter in the sequence appears more than once. For a complete mix, t is a sequence containing all shelters.

For n shelters, there are n! ways in which to arrange them ("permutations"). Given a permutation p of n shelters and client c_i, let z_i^{p} be the complete mix for c_i over the n shelters arranged by permutation p.

The third requirement for a PrivaMix function is that for all n! permutations of n shelters, $z_i^{\mathsf{p}} = z_i^{\mathsf{q}}$ where p and q are any two of the n! permutations of shelters.

The third requirement states that a PrivaMix function must be a commutative cipher[18]. For a given c_i, all complete mixes over the same n shelters yield the same value z_i regardless of the order in which the shelters mix. This is the "commutative" requirement.

The fourth requirement for a PrivaMix function is that client c_i cannot be learned even if u_{ij}, z_i and any sub-mixes z_i^{t} are shared.

According to the fourth requirement, a PrivaMix function must not reveal the secret client information even if sub-mixes are revealed. This is the "privacy" requirement.

The fifth requirement of a PrivaMix function is that given two UIDs, u_{ix} and u_{jy}, then $u_{ix}=u_{jy}$ if and only if i and j refer to the same client.

The fifth requirement states that the PrivaMix function must be collision free. It must be highly likely that if any two UIDs or complete mixes are the same, the originating clients are the same. This is the "collision free" requirement[19].

The sixth requirement for a PrivaMix function is that the complete mixes for all shelters visited by the same client must be the same. Given two complete mixes, z_i and z_j, then $z_i=z_j$ if and only if i and j refer to the same client.

The sixth requirement refers to the correctness of mixing results. For each shelter visited by a client, the complete mix for that client must be the same as those complete mixes of all other shelters visited by the same client. This is the "correctness" requirement[20].

[18] The cryptography community uses the term "commutative cipher" across multiple players to refer to a hashing or encrypting computation that provides the same value regardless of the order in which the hash or encrypted is performed.

[19] In order to satisfy the collision free requirement, the source information for a client must be unique and well chosen. This section, which describes the requirements for a PrivaMix function, assumes the source information, which is termed the client's secret value in this section, is unique and well chosen. More discussion about selecting appropriate client source information appears in Section 10.

[20] In order to satisfy the correctness requirement, the source information for a client must be reliably provided at each shelter visited. This section, which describes the requirements for a PrivaMix function, assumes the source information, is reliably provided at each shelter visited. More discussion about the reliability of client source information appears in Section 10.

The recommendation below summarizes the six requirements of a PrivaMix function.

*Recommendation #25: A PrivaMix function (**F**) must satisfy the following six requirements:*

1. *Inconsistent assignment: different shelters should generate different initial mix values for the same clients.*
2. *One-way function: **F** must be a one-way function.*
3. *Commutative: **F** must be a commutative cipher.*
4. *Privacy: the secret client information cannot be learned given the sharing of complete and sub-mixes.*
5. *Collision-free: mixes from **F** must be collision-free.*
6. *Correctness: all complete mixes for the same client must be the same. Complete mixes for different clients should not be the same.*

8.4 PrivaMix claims and limits

This section examines the appropriateness, correctness, and protection of PrivaMix. Discussion includes limitations, which do exist, though executing preliminary or additional secondary protocols, or adopting recognized best practices, as described, may improve restrictions. Before examining claims and limits, a summary of assumptions and threats appears.

Assumptions.

PrivaMix assumes Shelters are cooperative, non malicious participants that behave as instructed. The Planning Office is also a cooperative participant, but may attempt to learn private information during or after processing.

Review of Threats

Vulnerabilities appear when the Planning Office and/or a Shelter learns Client source information as a result of the existence or processing of UIDs. Only a Shelter that generates a UID for a Client should know the Client's source information.

PrivaMix de-duplicates UIDs while provably maintaining the privacy of Client source information. PrivaMix does not necessarily thwart data linkage attacks on the other Universal Data Elements, though a variation appeared earlier in Section 8.2 that provides an effective guard. As described in Section 4, a data linkage attack re-identifies Clients by matching combinations of values found in the Universal Data Elements to values found in other datasets. If matching is not based on UIDs, data linkage may lie outside the scope of the PrivaMix Protocol unless a variant is used to specifically address data linkage. The most effective way to thwart data linkage is to make sure the data elements of the client-level data cannot be linked to other readily available data. Section 11 revisits the identifiability of the Universal Data Elements in light of PrivaMix's protection for UIDs in order to make recommendations to thwart data linkage.

Here are the seven statements claimed about PrivaMix.

8.4.1. Usability claim. Communication time is linear in the number of Shelters.

8.4.2. Correctness claim. If the complete mixes are the same, the Clients representing the original UIDs presented the same source information..

8.4.3. Privacy claim. A dictionary attack by the Planning Office will not yield reliable re-identifications.

8.4.4. Privacy claim. Compromising a Shelter will not help the intimate stalker learn where a targeted Client is (or has been). Similarly, compromising the Planning Office will not help the intimate stalker learn where a targeted Client is (or has been).

8.4.5. Privacy claim. Even if the Planning Office pads the UIDs with known values, the Planning Office does not learn Client source information.

8.4.6. Limitation. If the Planning Office and at least one Shelter collude, the Planning Office can learn Client source information about the Shelter's Clients and the Shelter can learn other Shelters its Clients visited.

8.4.7. Limitation. If during the de-duplication protocol, the intimate stalker compromises both the Planning Office and a Shelter the targeted Client visited, the intimate stalker can learn the locations of all Shelters the Client visited. In addition, the Planning Office can learn the source information for that Client.

8.4.1. Usability claim. Communication time is linear in the number of Shelters.

Proof sketch:
In Step 3 of the PrivaMix Protocol (Section 8.2.1), the Planning Office sequentially requests each Shelter to mix all UIDs once.

The de-duplication step (Step 3) of the PrivaMix Protocol dictates the time it takes to execute PrivaMix. Adding a Shelter to the PrivaMix Network increases the time it takes to execute the PrivaMix Protocol because the additional Shelter will have to mix all UIDs once and mixing is done sequentially in the de-duplication step. The other steps can be done by Shelters in parallel.

Let v be the total number of Client visit records across m Shelters and t be the average time it takes a Shelter to mix the UIDs from all other Shelters, then the total time to execute the PrivaMix Protocol is on the order of ($v * t * m$). Therefore, time is linear in the number of Shelters (and visits).

Section 10 reports on a real-world experiment in which 4 Shelter computers work autonomously, without user intervention, to de-duplicate UIDs. It took about 45 minutes to de-duplicate UIDs for 2194 visits, which is less than 12 minutes per Shelter. In the real-world setting, there are 25 or fewer Shelter computers per Planning Office, so communication is practical.

8.4.2. <u>*Correctness claim.*</u> *If the complete mixes are the same, the Clients representing the original UIDs presented the same source information..*

> Proof sketch:
> This exploits the commutative property of the PrivaMix function. See Section 8.1 and Section 8.2 for examples and discussion.

8.4.3. <u>*Privacy claim.*</u> *A dictionary attack by the Planning Office will not yield reliable re-identifications.*

> Proof sketch:
> The size and nature of Shelter private values can be selected to ensure that exhaustively trying all possible combinations of private values over all possible values of source information is computationally infeasible.

Section 5.3 describes a dictionary attack in which the Planning Office tries all possible combination of values to see which values match those for whom the Client information is known. One example in Section 5.3 shows that encrypting Social Security numbers can be vulnerable to a dictionary attack when the Planning Office knows the encryption function. Within four seconds, the Planning Office can compute the UID for every possible Social Security number and then match UIDs from Shelters with those computed by the Planning Office to learn the Client's Social Security number.

PrivaMix protects against a dictionary attack by requiring the Planning Office to try all possible combinations of Shelter private values and Client Source information to learn Client source information. If Shelters chose sufficiently large private values, exhaustively attempting every combination is not feasible. See Figure 18.

Recommendations below help thwart dictionary attack possibilities.

<u>*Recommendation #26:*</u> *Each Shelter must select a sufficiently private value so that efforts by the Planning Office to exhaustively compute all combinations of Shelter private values and Client source information (a dictionary attack) are not feasible. Most likely a Shelter's computer will be required to select a private value 512 bits or larger (as appropriate and most likely randomly selected at the start of each reporting period).*

<u>*Recommendation #27:*</u> *To help thwart the possibility of the Planning Office or other Shelters learning a Shelter's private value, a Shelter may not even explicitly know its own private value for a reporting period –i.e., the computer program may generate it internally and not explicitly reveal it.*

<u>*Recommendation #28:*</u> *To help thwart the possibility of the Planning Office or other Shelters learning a Shelter's private value, a Shelter may make its private value available to its copy of the PrivaMix function only while mixing over the PrivaMix Network. Other parties should not be able to invoke a Shelter’s PrivaMix function with the Shelter's private value.*

8.4.4. <u>Privacy claim.</u> Compromising a Shelter will not help the intimate stalker learn where a targeted Client is (or has been). Similarly, compromising the Planning Office will not help the intimate stalker learn where a targeted Client is (or has been).

Even though the intimate stalker may know the Client's source information, he cannot relate UIDs across Shelters because he does not know the private values of Shelters not colluding with him. Therefore, his compromising a Shelter or the Planning Office to find a targeted individual will not be fruitful for the same reasons described above for the dictionary attack.

8.4.5. <u>Privacy claim.</u> Even if the Planning Office pads the UIDs with known values, the Planning Office does not learn Client source information.

Consider the case in which the Planning Office makes a set of UIDs using its own copy of the PrivaMix function and a private value it selects. During de-duplication, the Planning Office then merges its made-up UIDs with the UIDs from the other Shelters for mixing. Because the Planning Office knows the source information that it used to construct its UIDs, any matches with UIDs from Shelters reveals the source information of actual Clients at particular Shelters.

In order to combat this attack, Shelters run simple protocols to validate the number of UIDs and/or to mix UIDs without Planning Office involvement (see Variation 1 and Variation 2 in Section 8.2). There are many other possible ways to accomplish these protections.

<u>Recommendation #29:</u> In order to prevent the Planning Office from padding UIDs with known values, the original PrivaMix approach should be modified to validate the number of UIDs and/or to mix UIDs without Planning Office involvement (see Variation 1 and Variation 2 in Section 8.2).

8.4.6. <u>Limitation.</u> If the Planning Office and at least one Shelter collude, the Planning Office can learn Client source information about the Shelter's Clients and the Shelter can learn other Shelters its Clients visited.

One of the most likely ways this collusion happens is when one of the shelters in the PrivaMix Network is the regional HMIS, because in many geographical regions, the staff of the HMIS is the same staff as the Planning Office (or CoC) and because there is a desire to de-duplicate visits across the domestic violence homeless shelters and the HMIS (not the domestic violence homeless shelters alone). Unfortunately, if the HMIS and the Planning Office collude, presumably because they are the same staff, the Planning Office can learn Client identities and the HMIS can learn which Clients by name, visited which Shelters. This is not allowed under VAWA (Section 7).

What makes this a significant threat to Client re-identifications is that unlike the Planning Office colluding with another domestic violence homeless shelter, a HMIS will usually contain most (if not all) Clients who visit any domestic violence homeless shelter. When a Client of a domestic violence homeless shelter seeks services beyond the domestic violence homeless shelter (e..g., meals or medical care), her source information appears in the HMIS related to those services. The HMIS knows her source information and the general services she received, but the HMIS does not know she is a domestic violence homeless client or the domestic violence homeless shelter in which she resides. After de-duplication, the Planning Office learns the Client received services from a HMIS and from a domestic violence homeless shelter, and can use the identifying information in the HMIS to explicitly identify the Client.

In order to combat collusion between the HMIS and the Planning Office, the final de-duplicated information made available to the Planning Office by PrivaMix must be rendered unlinkable to the HMIS data that produced it in part. Remedies include having PrivaMix provide only aggregate information or provably anonymizing released data elements. Section 12 discusses these remedies in detail.

Recommendation #30: *Care must be taken to combat possible collusion between the HMIS and the Planning Office because in many geographical regions, the staff of the HMIS is the same staff as the Planning Office (or CoC) and because there is a desire to de-duplicate visits across the domestic violence homeless shelters and the HMIS (not the domestic violence homeless shelters alone). As a participant in PrivaMix, a HMIS poses a significant threat to Client re-identifications because a HMIS will usually contain most (if not all) Clients who visit any domestic violence homeless shelter. Remedies include having PrivaMix provide only aggregate information or provably anonymizing released data elements. Section 12 discusses these remedies in detail.*

8.4.7. Limitation. If during the de-duplication protocol, the intimate stalker compromises both the Planning Office and a Shelter the targeted Client visited, the intimate stalker can learn the locations of all Shelters the Client visited. In addition, the Planning Office can learn the source information for that Client.

The following two recommendations establish best practices to help.

Recommendation #31: Client records Shelters provide to the Planning Office should only include Clients who are no longer residing at the Shelter. This is a helpful recommendation, but not wholly satisfactory because Clients may re-visit previously visited Shelters.

Recommendation #32: The Planning Office should destroy all copies of the original UIDs once the de-duplication is complete. Doing so, limits the opportunity for compromise.

The claims and limits mentioned above reflect the generic PrivaMix approach. A specific implementation of a system that uses the PrivaMix approach requires revisiting claims and limits specific to implementation details. Differences in implementations may include communication flow (e.g. Planning Office in the middle or Shelter-to-Shelter), information content (e.g., a stream of values, or a list of values with their originating Shelter), and selection of the privately held Shelter value (.e.g., random selection, or pre-selection). Section 9 describes a particular instantiation of a PrivaMix system used in a real-world experiment (Section 10).

Recommendation #33: *A specific implementation of a system that uses the PrivaMix approach requires revisiting claims and limits specific to implementation details. Differences in implementations may include communication flow (e.g. Planning Office in the middle or Shelter-to-Shelter), information content (e.g., a stream of values, or a list of values with their originating Shelter), and selection of the privately held Shelter value (.e.g., random selection, or pre-selection).*

8.5 Comparison to other UID technologies

This section compares the PrivaMix approach with the UID technologies examined in Section 6 using the criteria for assessing the utility ("warranty") and privacy ("compliance") of UID technologies introduced in Section 5. PrivaMix performs comparable to inconsistent hashing (Section 6.7) and distributed query (Section 6.8) making it generally better than encoding (Section 6.1), hashing (Section 6.2), encryption (Section 6.3), scan cards and RFIDs (Section 6,4), biometrics (Section 6.5), and consent (Section 6.6) at protecting privacy. Yet, its usefulness at de-duplicating is better than encoding, hashing, encryption, scan cards and RFID, but not better than biometrics or consent. The following sections examine these statements in detail.

There are two main variants of PrivaMix to consider: one in which the result is de-duplicated Client-level visit information (Section 8.5.1) and one in which the result is de-duplicated aggregate count distributions (Section 8.5.2). These pose different ways of addressing data linkage threats. As described in Section 8.2, providing the Planning Office with aggregate count distributions gives a privacy guard within PrivaMix to help thwart data linkage. Examining the identifiability of Client-level data the Planning Office receives (Section 11), provides a privacy guard outside of PrivaMix. This section ends with an overall comparison of PrivaMix and other UID technologies (Section 8.5.3).

8.5.1. Assessment with client-level results

When the result from executing PrivaMix is Client-level data at the Planning Office, as described in Figure 55, then a gross assessment of PrivaMix as a UID technology yields warranty and compliance statements comparable to those of inconsistent hashing.

See Figure 63 and Figure 64 for a gross assessment of using PrivaMix as a UID technology when the result is client-level data. Issues related to utility and the warranty statement appear in Figure 63. Issues related to privacy and the compliance statement appear in Figure 64. While shadings may identify some problems as being of severe or moderate concern, depending on implementation details, these problems may be sufficiently addressed with straightforward practices, policies, or technology decisions.

PRIVAMIX (with Client-level results) –WARRANTY (UTILITY) STATEMENT

Non-Verifiable source information	■	*If a UID is based on non-verifiable source information provided by the Client that is not truthful or is inconsistently used, what happens?* Serious de-duplication problems are likely if Clients provide non-verifiable source information inconsistently. On the other hand, source information that is not truthful, but consistently provided, is typically not a problem.
Verifiable source information		*Can problems occur if the UID is based on verifiable source information?* Using invariant Client information that can be consistently verified on each visit is likely to avoid problems. Even if the information is not correct, but consistently verified on each visit, no problems are likely. An example of invariant verifiable Client information is a reliably captured biometric.
Client confidence and trustworthiness		*How trustworthy is the UID likely to be perceived by Clients (as well as by those who regularly intake Clients)?* Like hashed UIDs, PrivaMix UIDs tend to appear cryptic, which can instill Client and intaker confidence and thereby avoid problems. Further, because UIDs are different across Shelters (and can even be different on multiple visits to the same Shelter), additional Client and intaker confidence can be attained. Problems may emerge based on the sensitivity of requested source information despite the cryptic appearance of the UID itself, bu in most PrivaMix implementations the UIDs are not ever visible.
Inflated accounting	■	*What are the circumstances under which de-duplication is likely to inflate the accounting?* Count inflation can occur in cases where a Client provides different source information on different visits. In these cases, different UIDs are generated and therefore will not match to each other even though they are assigned to the same Client. Count inflation can also occur in cases in which a Client provides incomplete or missing information on different visits, thereby producing different non-matchable UIDs across Shelters.
Deflated accounting	▒	*What are the circumstances under which de-duplication is likely to deflate the accounting?* Count deflation is possible when different Clients provide identical complete and incomplete information. A glaring example occurs for Clients in which all relevant source information is missing. Attention should be paid to how these situations are addressed in UIDs across Shelters. Count inflation is more likely than deflation.

…continued on next page …

Handling bad or missing input	■	*What is the effect of bad, incomplete, or missing source information on performance?* Typing mistakes and incomplete or missing information can generate different UIDs for a Client than would have been generated with complete and properly entered information. This tends to inflate accounting by generating spurious UIDs for Clients having multiple visits. Incomplete and missing information may also inflate accounting. Inflation is more likely than deflation.

■	Most severe/difficult problem
▓	Moderate problem
▓	A problem
░	May be a problem
□	No problem likely, or not applicable

Figure 63. Gross Warranty assessment of using PrivaMix (with Client-level results) as a UID technology.

PRIVAMIX (with Client-level results) –COMPLIANCE (PRIVACY) STATEMENT

Intimate Stalker	░	*What vulnerabilities exist for the intimate stalker?* Because each Shelter has a different UID for the same Client, access to Shelter information is limited to a Shelter-by-Shelter basis. Vulnerabilities that are able to be exploited by an intimate stalker are limited to the Planning Office, which controls the de-deduplicated result. Vulnerabilities at the Planning Office may be addressed by the selection of data elements that comprise the de-duplicated results, and by control and audit of de-duplicated results.
Re-identification: Linking	░	*What vulnerabilities exist for re-identification of UIDs (and Dataset) using data linkage on UIDs?* Because a different UID is generated at each Shelter a Client visits, and the UIDs are not used outside HMIS data, unauthorized linking on UIDs is not likely. Re-identification not using UIDs is possible. Remedies rely on anonymizing data values (Section 8.2.6) or data elements (Section 11).
Re-identification: Dictionary Attack		*What vulnerabilities exist for re-identification of UIDs (and Dataset) using a dictionary attack on UIDs?* Because there should be a large range of possible UID values, a different UID generated at each Shelter a Client visits, and the non-use of UIDs used outside de-deuplication, a dictionary attack is not likely to be fruitful because of the large number of possibilities. However, care must be taken to make sure that no additional UIDs are added to the mix by the Planning Office. See PrivaMix variation (Section 8.2.3) for a remedy.
Re-identification: Reversal		*What is involved in reverse engineering the UID construction method?* Because PrivaMix functions are strong, reversal is not usually an issue. But if a Shelters' PrivaMix function and private value are available to unlimited use by the Planning Office, re-identification can result. Care must be taken to control or limit the function's use to avoid unwanted dictionary attacks (discussed above) or reverse compilations. (A dictionary is more likely than an attempt to reverse compile the function.)
Exposure		*What legal or technical risks or liabilities may be introduced based on the existence of the resulting database or UID technology?* The existence of mixed UIDs used only in the HMIS-context is not likely to expose Clients to additional risks beyond those mentioned above.

■	Most severe/difficult problem
▓	Moderate problem
▓	A problem
░	May be a problem
□	No problem likely, or not applicable

System Trust *Which parties are heavily trusted?* Planning Offices are heavily trusted to control access and use of results.

Figure 64. Gross Compliance assessment of using PrivaMix (with Client-level results) as a UID technology.

8.5.2. Assessment with aggregate results

Rather than PrivaMix providing Client-level data, as described in Figure 55, the result can be the AHAR report itself or some other representation of aggregate count distributions. When the result is aggregate count distributions, a gross assessment of PrivaMix as a UID technology yields the same utility (or warranty), as shown in Figure 65, but improved privacy (or compliance), as shown Figure 66, than results with Client-level data.

Issues related to utility and the warranty statement appear in Figure 65. Issues related to privacy and the compliance statement appear in Figure 66. While shadings may identify some problems as being of severe or moderate concern, depending on implementation details, these problems may be sufficiently addressed with straightforward practices, policies, or technology decisions.

PRIVAMIX (with aggregate results) –WARRANTY (UTILITY) STATEMENT

Non-Verifiable source information	■	*If a UID is based on non-verifiable source information provided by the Client that is not truthful or is inconsistently used, what happens?* Serious de-duplication problems are likely if Clients provide non-verifiable source information inconsistently. On the other hand, source information that is not truthful, but consistently provided, is typically not a problem.
Verifiable source information		*Can problems occur if the UID is based on verifiable source information?* Using invariant Client information that can be consistently verified on each visit is likely to avoid problems. Even if the information is not correct, but consistently verified on each visit, no problems are likely. An example of invariant verifiable Client information is a reliably captured biometric.
Client confidence and trustworthiness		*How trustworthy is the UID likely to be perceived by Clients (as well as by those who regularly intake Clients)?* Releasing only aggregate count results can instill Client and intaker confidence and avoid problems, especially since aggregate results are based on PrivaMix UIDs that appear cryptic, and are different across Shelters (and can even be different on multiple visits to the same Shelter). Problems may still emerge based on the sensitivity of requested source information. Educating Clients and those who perform intake regularly and/or issuing privacy notices may help.
Inflated accounting	■	*What are the circumstances under which de-duplication is likely to inflate the accounting?* Count inflation can occur in cases where a Client provides different source information on different visits. In these cases, different UIDs are generated and therefore will not match to each other even though they are assigned to the same Client. Count inflation can also occur in cases in which a Client provides incomplete or missing information on different visits, thereby producing different non-matchable UIDs across Shelters.
Deflated accounting	▒	*What are the circumstances under which de-duplication is likely to deflate the accounting?* Count deflation is possible when different Clients provide identical complete and incomplete information. A glaring example occurs for Clients in which all relevant source information is missing. Attention should be paid to how these situations are addressed in UIDs across Shelters. Count inflation is more likely than deflation.

…continued on next page …

Handling bad or missing input	■	*What is the effect of bad, incomplete, or missing source information on performance?* Typing mistakes and incomplete or missing information can generate different UIDs for a Client than would have been generated with complete and properly entered information. This tends to inflate accounting by generating spurious UIDs for Clients having multiple visits. Incomplete and missing information may also inflate accounting. Inflation is more likely than deflation.

■	Most severe/difficult problem
▓	Moderate problem
▓	A problem
░	May be a problem
□	No problem likely, or not applicable

Figure 65. Gross Warranty assessment of using PrivaMix (with aggregate results) as a UID technology.

PRIVAMIX (with aggregate results) –COMPLIANCE (PRIVACY) STATEMENT

Intimate Stalker		*What vulnerabilities exist for the intimate stalker?* Because each Shelter has a different UID for the same Client, access to Shelter information is limited to a Shelter-by-Shelter basis. Vulnerabilities related to exploiting the Planning Office are very limited since only aggregate count information is available. Care should be taken for the Planning Office not to even save UIDs and mixed UIDs.
Re-identification: Linking		*What vulnerabilities exist for re-identification of UIDs (and Dataset) using data linkage on UIDs?* If only aggregate count information results, linkage on UIDs and the Dataset is limited. Vulnerabilities at the Planning Office may be further minimized by the system not releasing mixed or non-mixed UIDs.
Re-identification: Dictionary Attack		*What vulnerabilities exist for re-identification of UIDs (and Dataset) using a dictionary attack on UIDs?* Because there should be a large range of possible UID values, a different UID generated at each Shelter a Client visits, and the non-use of UIDs used outside de-deuplication, a dictionary attack is not likely to be fruitful because of the large number of possibilities. However, care must be taken to make sure that no additional UIDs are added to the mix by the Planning Office. Section 8.2.3 poses a remedy.
Re-identification: Reversal		*What is involved in reverse engineering the UID construction method?* Because PrivaMix functions are strong, reversal is not usually an issue. But if a Shelters' PrivaMix function and private value are available to unlimited use by the Planning Office, re-identification can result. Care must be taken to control or limit the function's use to avoid unwanted dictionary attacks (discussed above) or reverse compilations. (A dictionary is more likely than an attempt to reverse compile the function.)
Exposure		*What legal or technical risks or liabilities may be introduced based on the existence of the resulting database or UID technology?* The existence of mixed UIDs used only in the HMIS-context is not likely to expose Clients to additional risks beyond those mentioned above.

■	Most severe/difficult problem
▓	Moderate problem
▓	A problem
░	May be a problem
□	No problem likely, or not applicable

System Trust
Which parties are heavily trusted?

Planning Offices are trusted to control access and use of results, but with aggregate results only, sharing concerns are minimal.

Figure 66. Gross Compliance assessment of using PrivaMix (with aggregate results) as a UID technology.

8.5.3. Overall comparison

Figure 67 compares PrivaMix , with Client-level and aggregate count distributions, with the UID technologies examined in Section 6. PrivaMix performs comparable to inconsistent hashing (Section 6.7) and distributed query (Section 6.8) making it generally better than encoding (Section 6.1), hashing (Section 6.2), encryption (Section 6.3), scan cards and RFIDs (Section 6,4), biometrics (Section 6.5), and consent (Section 6.6) at protecting privacy. Yet, the utility of its de-duplicated results is better than encoding, hashing, encryption, scan cards and RFID, but not better than biometrics or consent.

	UTILITY						PRIVACY				
UID TECHNOLOGY	Non-verifiable source	Verifiable source	Client Trust	Inflate Accounting	Deflate Accounting	Bad or missing info	Intimate stalker	Linking	Dictionary attack	Reverse engineer	Expose new issues
Encoding	■	░	▨	░	■	■	■	■	■	■	▨
Hashing	■	□	░	░	■	■	■	■	■	▨	□
Encryption	■	□	▨	░	■	■	■	■	■	▨	▨
Scan Cards/RFID	▨	░	■	■	░	░	▨	▨	░	░	■
Biometrics	□	□	░	▨	▨	░	▨	▨	▨	▨	■
Consent	■	□	░	▨	▨	░	■	■	■	■	■
Inconsistent Hash	■	□	░	■	░	■	▨	□	▨	░	□
Distributed Query	■	□	□	■	░	■	░	□	□	□	□

PrivaMix (client-level)	■	□	□	■	░	■	░	░	□	□	□
PrivaMix (aggregate)	■	□	□	■	░	■	□	□	□	□	□

■	Most severe/difficult problem
▨	Moderate problem
▨	A problem
░	May be a problem
□	No problem likely, or not applicable

Figure 67. Summary of gross assessments of UID technologies,including PrivaMix variants.

9. The PrivaMix Demonstration System, as used in Iowa

In 2007, Privacert implemented a version of the PrivaMix Protocol (Section 8.2) for a real-world experiment; we term this software the "PrivaMix Demonstration System" (or merely "System"). Because there are numerous variations and many ways to implement the PrivaMix Protocol, this section describes the details of the PrivaMix Demonstration System specifically. Section 10 explains its use in the real-world experiment.

In the PrivaMix Demonstration System, each participating machine runs special software devoted to this task. Shelter machines run one edition of the software program ("the Shelter Edition"). The Planning Office machine runs a different edition ("the CoC Edition"). These editions differ because the responsibilities of Shelters and the Planning Office in the PrivaMix protocol are different.

Appendix A contains a copy of the Software User's Guide for the PrivaMix Demonstration System. It itemizes menu options and shows copies of screen shots at various points in operation. Overall, the operation is extremely simple. If Shelters and the Planning Office use default settings, then operation is as simple as loading the Client information and clicking one button.

A description of the following characteristics further describe an implementation of PrivaMix. Each appears as a subsection below.

9.1 Hardware and network assumptions
9.2 The PrivaMix function
9.3 Selection and size of shelter private values
9.4 Selection and size of Client source information
9.5 Transfer of Universal Data Elements
9.6 UID validation
9.7 De-duplication network
9.8 Post processing
9.9 Comparison to prior recommendations

9.1 Hardware and network assumptions

The PrivaMix Demonstration System has minimal machine requirements, which means almost any computer system sold today is sufficient for use. However, the machine must have access to the Internet. Below is more information about these requirements.

9.1.1. Using the Internet

Machines participating in the PrivaMix Demonstration System communicate using the Internet Protocol on the Internet through traditional means of accessing the Internet. The software encrypts all communications between Shelters and the Planning Office.

The PrivaMix Demonstration System works with all traditional forms of Internet connections, even wireless broadband access. In wireless broadband access, a special card fits into the computer. The card then communicates directly with a wireless mobile phone network to send and receive information over the Internet. Wireless broadband is usually slower than dial-up, where a machine uses a phone line to communicate over the Internet, and is usually slower than Cable Internet, where a network cable connects directly to the computer. Just as the PrivaMix Demonstration System works with wireless broadband, it also works with dial-up, and cable connections. In fact, participants can use a mixture of Internet connection methods.

In general, communication in the PrivaMix Demonstration System consists of transmitting information bundles between a Shelter and its Planning Office. For example, Shelters send Client visit information to the Planning Office as an information bundle. The Planning Office sends UIDs and mixes to Shelters for mixing as an information bundle of one or more UIDs at a time. And, Shelters return mix results to the Planning Office as an information bundle. Communication is therefore more episodic in nature than continuous. All these communications use the Internet Protocol.

In the Internet Protocol, each machine on the Internet has its own unique number; this is termed the machine's “IP address.” We assume that the IP address of a machine in the PrivaMix Demonstration System is unknown at software start. To learn the IP addresses of participating machines, each participating machines accesses a special program running on a previously known server which gathers and then reports relevant IP addresses among participants. As a result of connecting to the server, each participating Shelter learns the IP address of the Planning Office and the Planning Office learns the IP addresses of all participating Shelters. No other communication is made with the server unless a Shelter machine restarts during the PrivaMix Protocol and is assigned a different IP address.

9.1.2. Machine requirements

The PrivaMix Demonstration System has minimal machine requirements given the typical configuration of today's machines. As described above, the machine must have an Internet connection. It must have enough hard drive space to store the Client visit information. Standard machine configurations for memory and processing power are sufficient. The software works with popular operating systems (Windows, Linux, Mac OS).

9.1.3. Machine and network security

The PrivaMix Demonstration System assumes the machine is able to function properly without viruses or other impediments. No additional security requirements exist beyond the accepted security practices for maintaining Client information.

9.2 The PrivaMix function

The PrivaMix Demonstration System uses a strong one-way function. For details about the function and its proofs of correctness and compliance to the six requirements of a PrivaMix function, see [32]. Here are two key features. The function includes the modulus operator, so it is not easily reversed. The modulus operator is embedded within the function so that it maintains the commutative property across Shelters. See [32] for details and descriptions of this and other possible PrivaMix functions.

9.3 Selection and size of Shelter private values

The PrivaMix Demonstration System automatically selects a random 64-bit value as the Shelter's private value. The System selects the value after the Shelter provides Client source information. At that point, the System must generates UIDs, which requires its use. The System never reveals the Shelter's private value to the Shelter or the Planning Office. The value resides only in the Shelter's RAM memory. It is not stored or shared. If the machine has to restart during use, the System will select another value for the Shelter's private value and the network will start mixing again.

While the System uses 64 bit values, this is an internal setting. The PrivaMix Demonstration System supports 32, 64, 128, and 256 bit values.

9.4 Selection and size of Client source information

The PrivaMix Demonstration System does not prescribe which fields to use as Client source information. Given a set of fields identified for use, the System will compute a 64-bit UID for the Client. Just as with private values for Shelters, the PrivaMix Demonstration System uses 64 bit values for resulting UIDs. This is an internal setting. The PrivaMix Demonstration System also supports 32, 64, 128, and 256 bit values.

While the PrivaMix Demonstration System does not dictate which Client fields to use as source information, precautions are needed. Here are two important precautions.

(7) Care must be taken that sufficient variability exists in the fields so that resulting UIDs have a sufficiently wide range of possible values.

(8) Care must also be taken to make sure that different Clients are not likely to have to the same set of values appearing in the source information.

Further details about selecting Client source information to best work with a PrivaMix function appears in [32].

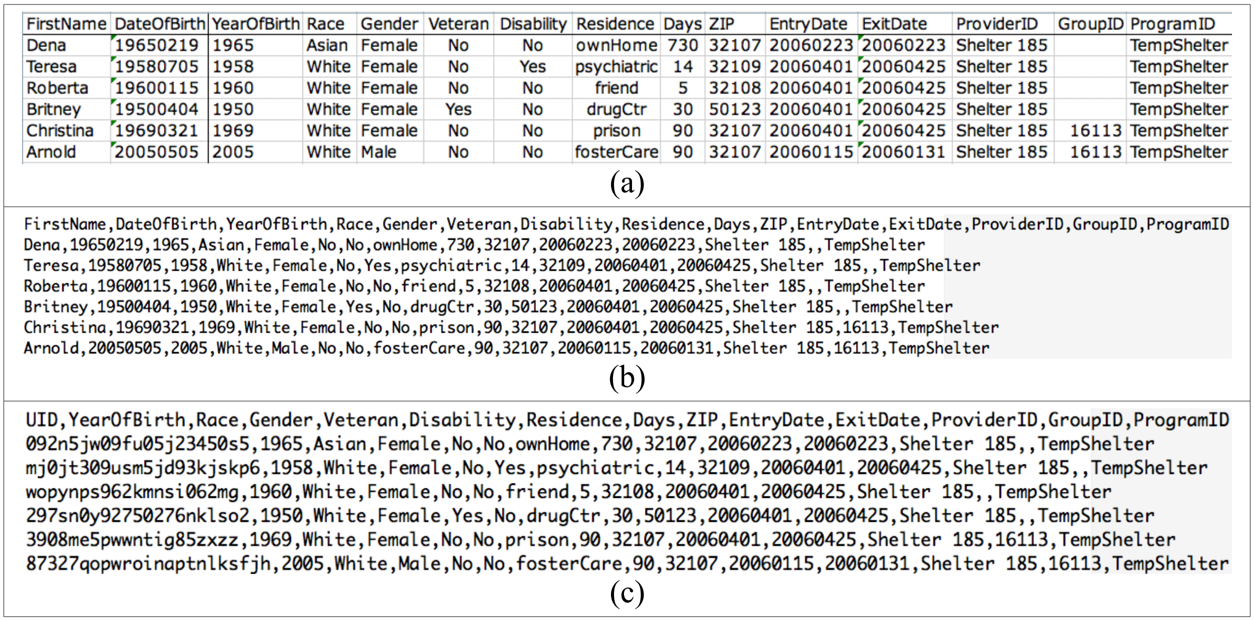
9.5 Transfer of Universal Data Elements

In the PrivaMix Demonstration System, Shelters transfer a comma-delimited text file to the Planning Office as encrypted content over an Internet connection. Each line contains a Client's visit information. The leftmost field on the line is the Client's UID. The remaining fields on the line are fields associated with the Client's visit to the Shelter, presumably the Universal Data Elements associated with that Client.. Below are more details about the file.

A comma-delimited text file is a simple text file that stores a table of information as follows. Each row of the table is a line in the text file. Columns in the table appear in order, left to right, with values separated by commas. The values themselves may be enclosed in quotation marks. Figure 68 shows an example of a comma-delimited text file. The original table shown in Figure 68(a) appears as comma-delimited file in Figure 68(b). A comma-delimited text file can be composed in a word processor (e.g., Microsoft Word), in a text editor (e.g., Notepad), or converted from a spreadsheet (e.g., Excel) or database program (e.g. Access).

In the PrivaMix Demonstration System, two versions of a comma-delimited text file exist. The Shelter provides an initial comma-delimited text file for processing; see Figure 68(b). This text file has the fields that comprise the Client's source information appearing as the leftmost fields. The fields that are to be sent to the Planning Office associated with that Client appear in the remaining fields. After a Shelter machine computes UIDs for each of its Clients using the Client source information, it produces a comma-delimited file replacing the leftmost fields with the Client UID; see Figure 68(c). Copies of UID information appear in temporary memory only. No copies appear on the hard drive.[21] Processing only involves the fields of the Client source information. The Planning Office receives the other fields with no processing or review. The Shelter machine merely forwards them “as is” to the Planning Office.

[21] In the PrivaMix Demonstration System, copies of information containing UIDs appear only in the computer's memory (RAM). No copies appear on the machine's hard drive. However, if the size of the files warranted more storage than available in the computer's memory, encrypted copies could appear on the hard drive without posing security concern.



FirstName	DateOfBirth	YearOfBirth	Race	Gender	Veteran	Disability	Residence	Days	ZIP	EntryDate	ExitDate	ProviderID	GroupID	ProgramID
Dena	19650219	1965	Asian	Female	No	No	ownHome	730	32107	20060223	20060223	Shelter 185		TempShelter
Teresa	19580705	1958	White	Female	No	Yes	psychiatric	14	32109	20060401	20060425	Shelter 185		TempShelter
Roberta	19600115	1960	White	Female	No	No	friend	5	32108	20060401	20060425	Shelter 185		TempShelter
Britney	19500404	1950	White	Female	Yes	No	drugCtr	30	50123	20060401	20060425	Shelter 185		TempShelter
Christina	19690321	1969	White	Female	No	No	prison	90	32107	20060401	20060425	Shelter 185	16113	TempShelter
Arnold	20050505	2005	White	Male	No	No	fosterCare	90	32107	20060115	20060131	Shelter 185	16113	TempShelter

(a)

```
FirstName,DateOfBirth,YearOfBirth,Race,Gender,Veteran,Disability,Residence,Days,ZIP,EntryDate,ExitDate,ProviderID,GroupID,ProgramID
Dena,19650219,1965,Asian,Female,No,No,ownHome,730,32107,20060223,20060223,Shelter 185,,TempShelter
Teresa,19580705,1958,White,Female,No,Yes,psychiatric,14,32109,20060401,20060425,Shelter 185,,TempShelter
Roberta,19600115,1960,White,Female,No,No,friend,5,32108,20060401,20060425,Shelter 185,,TempShelter
Britney,19500404,1950,White,Female,Yes,No,drugCtr,30,50123,20060401,20060425,Shelter 185,,TempShelter
Christina,19690321,1969,White,Female,No,No,prison,90,32107,20060401,20060425,Shelter 185,16113,TempShelter
Arnold,20050505,2005,White,Male,No,No,fosterCare,90,32107,20060115,20060131,Shelter 185,16113,TempShelter
```

(b)

```
UID,YearOfBirth,Race,Gender,Veteran,Disability,Residence,Days,ZIP,EntryDate,ExitDate,ProviderID,GroupID,ProgramID
092n5jw09fu05j23450s5,1965,Asian,Female,No,No,ownHome,730,32107,20060223,20060223,Shelter 185,,TempShelter
mj0jt309usm5jd93kjskp6,1958,White,Female,No,Yes,psychiatric,14,32109,20060401,20060425,Shelter 185,,TempShelter
wopynps962kmnsi062mg,1960,White,Female,No,No,friend,5,32108,20060401,20060425,Shelter 185,,TempShelter
297sn0y92750276nklso2,1950,White,Female,Yes,No,drugCtr,30,50123,20060401,20060425,Shelter 185,,TempShelter
3908me5pwwntig85zxzz,1969,White,Female,No,No,prison,90,32107,20060401,20060425,Shelter 185,16113,TempShelter
87327qopwroinaptnlksfjh,2005,White,Male,No,No,fosterCare,90,32107,20060115,20060131,Shelter 185,16113,TempShelter
```

(c)

Figure 68. Comma-delimited text file having Client visit information. Original table (a) having 6 records and 14 fields appears as an equivalent comma-delimited file (b). The first line of the file includes the list of field names. The two leftmost fields, FirstName and DateOfBirth, are Client source information. The remaining fields provide Client visit information (see Figure 5 for descriptions). In the comma-delimited file in (c), the Client source information fields are replaced with a UID field. All other values remain the same. Dates appear as year, month, day (yyyymmdd). ProvideID is the Shelter's ID number. GroupID identifies Clients belonging to the same household. ProgramID identifies the kind of service provided.

9.6 UID validation

While Section 8.2.3 describes a variation of PrivaMix in which Shelters validate the number of values the Planning Office asks them to mix, the PrivaMix Demonstration System makes no such check. Shelter machines automatically mix values provided by the Planning Office without counting how many values that may be. This leaves a vulnerability: if the Planning Office pads the UIDs with known values, the Planning Office could learn Client source information (see Section 8.4.5). A simple remedy appears in Section 8.2.3, but in the interest of available resources, the PrivaMix Demonstration did not implement this variation.

9.7 De-duplication network

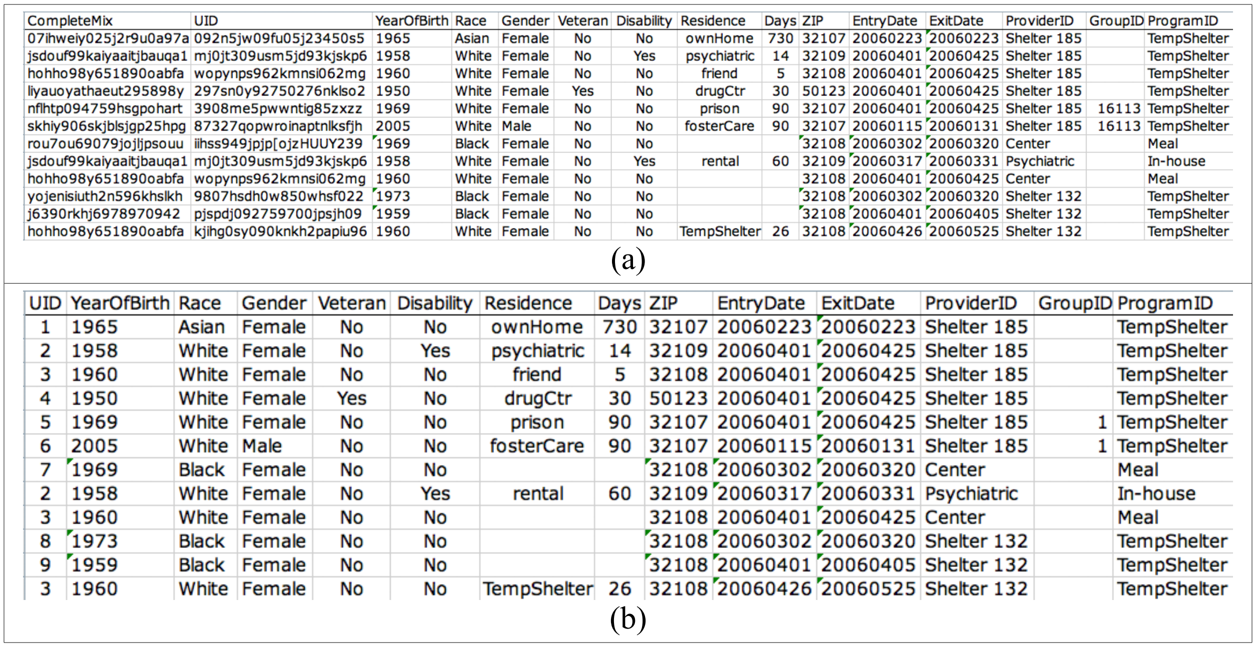
In the PrivaMix Demonstration System, the Planning Office orchestrates mixing as described in the generic PrivaMix Protocol (Section 8.2). The Planning Office sends values to each Shelter, one Shelter at a time, to mix, such that each Shelter mixes each UID once and each Shelter mixes all UIDs. Each Shelter only responds to mixing requests from the Planning Office's machine.

After mixing completes, the PrivaMix Demonstration System performs de-duplication on the Planning Office machine matching complete mixes across Shelter data. All values are held in the computer's memory. No information appears on the hard drive.

9.8 Post processing

Before making final de-duplicated results available to the Planning Office, the PrivaMix Demonstration System removes all UIDs, replacing them with numbers from 1 to the total number of distinct Clients and can do similar processing on PINs and Household IDs. The Planning Office does not receive a copy of the UIDs or complete mixes, only the results of de-duplication. Figure 68 shows the kinds of results made available to the Planning Office.

The PrivaMix Demonstration System automatically replaces UIDs with serialized numbers. An option exists by which other fields can be identified for serial renumbering. For example, in Figure 69, the Group ID, which identifies persons belonging to the same households, is serially renumbered.



CompleteMix	UID	YearOfBirth	Race	Gender	Veteran	Disability	Residence	Days	ZIP	EntryDate	ExitDate	ProviderID	GroupID	ProgramID
07ihweiy025j2r9u0a97a	092n5jw09fu05j23450s5	1965	Asian	Female	No	No	ownHome	730	32107	20060223	20060223	Shelter 185		TempShelter
jsdouf99kaiyaaitjbauqa1	mj0jt309usm5jd93kjskp6	1958	White	Female	No	Yes	psychiatric	14	32109	20060401	20060425	Shelter 185		TempShelter
hohho98y651890oabfa	wopynps962kmnsi062mg	1960	White	Female	No	No	friend	5	32108	20060401	20060425	Shelter 185		TempShelter
liyauoyathaeut295898y	297sn0y92750276nklso2	1950	White	Female	Yes	No	drugCtr	30	50123	20060401	20060425	Shelter 185		TempShelter
nflhtp094759hsgpohart	3908me5pwwntig85zxzz	1969	White	Female	No	No	prison	90	32107	20060401	20060425	Shelter 185	16113	TempShelter
skhiy906skjblsjgp25hpg	87327qopwroinaptnlksfjh	2005	White	Male	No	No	fosterCare	90	32107	20060115	20060131	Shelter 185	16113	TempShelter
rou7ou69079jojljpsouu	iihss949jpjp[ojzHUUY239	1969	Black	Female	No	No			32108	20060302	20060320	Center		Meal
jsdouf99kaiyaaitjbauqa1	mj0jt309usm5jd93kjskp6	1958	White	Female	No	Yes	rental	60	32109	20060317	20060331	Psychiatric		In-house
hohho98y651890oabfa	wopynps962kmnsi062mg	1960	White	Female	No	No			32108	20060401	20060425	Center		Meal
yojenisiuth2n596khslkh	9807hsdh0w850whsf022	1973	Black	Female	No	No			32108	20060302	20060320	Shelter 132		TempShelter
j6390rkhj6978970942	pjspdj092759700jpsjh09	1959	Black	Female	No	No			32108	20060401	20060405	Shelter 132		TempShelter
hohho98y651890oabfa	kjihg0sy090knkh2papiu96	1960	White	Female	No	No	TempShelter	26	32108	20060426	20060525	Shelter 132		TempShelter

(a)

UID	YearOfBirth	Race	Gender	Veteran	Disability	Residence	Days	ZIP	EntryDate	ExitDate	ProviderID	GroupID	ProgramID
1	1965	Asian	Female	No	No	ownHome	730	32107	20060223	20060223	Shelter 185		TempShelter
2	1958	White	Female	No	Yes	psychiatric	14	32109	20060401	20060425	Shelter 185		TempShelter
3	1960	White	Female	No	No	friend	5	32108	20060401	20060425	Shelter 185		TempShelter
4	1950	White	Female	Yes	No	drugCtr	30	50123	20060401	20060425	Shelter 185		TempShelter
5	1969	White	Female	No	No	prison	90	32107	20060401	20060425	Shelter 185	1	TempShelter
6	2005	White	Male	No	No	fosterCare	90	32107	20060115	20060131	Shelter 185	1	TempShelter
7	1969	Black	Female	No	No			32108	20060302	20060320	Center		Meal
2	1958	White	Female	No	Yes	rental	60	32109	20060317	20060331	Psychiatric		In-house
3	1960	White	Female	No	No			32108	20060401	20060425	Center		Meal
8	1973	Black	Female	No	No			32108	20060302	20060320	Shelter 132		TempShelter
9	1959	Black	Female	No	No			32108	20060401	20060405	Shelter 132		TempShelter
3	1960	White	Female	No	No	TempShelter	26	32108	20060426	20060525	Shelter 132		TempShelter

(b)

Figure 69. De-duplicated results. Information in the computer's memory after mixing (a). In the next step, the PrivaMix Demonstration System matches complete mixes to identify which clients are the same clients (rows 2 and 8; and, rows 3, 9, and 12). Planning Office receives a copy of de-duplicated results (b) with all UIDs replaced with numbering from 1 to the number of distinct Clients, repeating numbers to identify which records relate to the same Clients. Dates appear as year, month, day (yyyymmdd). GroupID identifies Clients belonging to the same household; these are also renumbered. Client 2 visited Shelter 185 and a psychiatric facility. Client 3 visited two Shelters and received meals.

9.9 Comparison to prior recommendations

Figure 70 below is a summary of the PrivaMix Demonstration System in terms of prior recommendations.

1	Outside scope.	Coordination of Systems across neighboring CoC's.
2	Outside scope.	Not share Shelter PIN beyond Shelter.
3	Implemented. (Section 9.8)	De-duplicated results should not include PINs, UIDs, or Household IDs.
4	Outside scope.	Shelters only include Clients who have left the Shelter.
5	Outside scope.	Train personnel on accepted practices for handling Client data.
6	Implemented. (Section 8.2)	UIDs should be inconsistently assigned across Shelters.
7	Outside scope.	Shelters should privacy notices for Client inspection.
8	Outside scope.	Fields date of birth and ZIP should be less specific.
9	Outside scope.	Planning Office should delete any fields in the Universal Data Elements not needed.
10	Outside scope.	Planning Office should sign Data Use Agreement with Shelters regarding linking.
11	Implemented. (Section 8.4)	Skilled person should certify System's risk of re-identification.
12	Implemented. (Section 8.4)	Skilled person should certify utility of de-duplicated results.
13	Implemented. (Section 8.4)	System using non-verifiable source information should instill trust.
14	Implemented. (Section 9.2)	System using encryption or hashing should use strong cryptographic methods.
15	Implemented. (Section 9.3)	System using encryption or hashing should control access to the function.
16	Outside scope.	System using scan cards/RFID should avoid issuing multiple cards to the same Client.
17	Implemented. (Section 9.8)	UIDs should be removed from de-duplicated results.
18	Outside scope.	Fields date of birth and ZIP must be less specific.
19	Implemented. (Section 8.4)	System must satisfy VAWA's requirements limiting re-idenification.
20	Implemented. (Section 9)	A PrivaMix System must avoid Shelters producing the same UID for Clients.
21	Addressed .(Section 9.1)	Computers transmitting UDE over a network must adhere to accepted security standards.
22	Not done. (Section 9.7)	If desirable, have a party other than the Planning Office orchestrate mixing.
23	Not done. (Section 9.8)	A PrivaMix System should anonymize or aggregate, rather than provide Client-level data.
24	Not done. (Section 9)	An economical PrivaMix System can result from using existing web browsers.
25	Implemented. (Section 9.2)	A PrivaMix Function must satisfy six noted requirements.
26	Implemented. (Section 9.3)	In a PrivaMix System. A Shelter value must be sufficiently large.
27	Implemented. (Section 9)	In a PrivaMix System, a Shelter should not even know its own private value.
28	Implemented. (Section 9.7)	In a PrivaMix System, unauthorized parties should be unable to use a Shelter's PrivaMix function.
29	Not done. (Section 9.6)	In a PrivaMix System, Shelters should validate the number of UIDs requested to mix.
30	Not done. (Section 9.8)	In order to provide collusion with an HMIS, provide only aggregate or anonymized results.
31	Outside scope.	Shelters only include Clients who have left the Shelter.
32	Implemented. (Section 9.8)	UIDs should be removed from de-duplicated results.
33	Implemented. (Section 9)	Claims must be assessed for any particular PrivaMix implementation.

Figure 70. Assessment of the PrivaMix Demonstration System in terms of prior recommendations made.

10. The Iowa Experiment

On June 6, 2007, a Planning Office in Iowa, along with three community Shelters and the area's HMIS tested the PrivaMix Demonstration System in three experiments. Jointly, we term these “the Iowa Experiment.” One experiment concerned the uniqueness of Client source information that did not use Social Security numbers. One experiment used a network of computers to test the ability of the software to de-duplicate. The final experiment examined the identifiability of the de-duplicated results. Below are details of these experiments.

10.1 Materials

Below is a description of the materials used in the Iowa Experiment.

10.1.1. Computers

The Iowa Experiment used five laptops in their original factory configurations. Four of the machines were Toshiba Satellite M115 laptops. Each Toshiba machine had an Intel Celeron M at 1.6GHz processor running the Windows XP operating system, 448MB of RAM memory, and 74GB of hard drive space. Among standard ports, each machine included a PCMCIA port. The original cost was about $500 each.

The Dell laptop had an Intel Centrino Duo (two processors) at 1.83Ghz, running the Windows XP operating system, 2 GB of RAM memory, and 93GB of hard drive space. Among standard ports, this machine included a PCMCIA port. The original cost was about $2000. The Dell laptop was significantly more powerful that the Toshiba machines.

These machines do not reflect the minimum machine requirements, as much as a description of the actual machines used. By providing standard laptops rather than using machines already at Shelters, these experiments were able to focus on performance issues rather than software installation and other secondary problems that can emerge in attempting to load software on unknown machines.

10.1.2. Network

Even though each laptop had all the standard Internet connection options (modem, wireless Internet, and Ethernet) built-in, the Iowa Experiment used five wireless broadband cards (4 Verizon and 1 Sprint), one per machine. The Verizon cards used the PCMCIA slots on the laptops. The Sprint card used the USB port.

A wireless broadband card communicates directly with a wireless mobile phone network to send and receive information over the Internet. This is usually slower than dial-up or cable Internet options.

The PrivaMix Demonstration System does not require the use of wireless broadband access to the Internet. By using these cards in standard laptops, the experiments did not have to assume participants were technically able to provide Internet access to the laptops.

Together, the laptops and network cards provided standardized hardware so that the experiments focus efficiently and narrowly on de-duplication performance.

10.1.3. PrivaMix Demonstration System

Each laptop ran an edition of PrivaMix Demonstration System (version 0.36). One machine designated as the Planning Office machine ran the CoC edition. The other four machines ran the Shelter Edition. Section 9 contains a detailed description of the PrivaMix Demonstration System. Appendix A has a copy of the User's Guide.

10.2 Subjects

Subjects are clients whose data appeared at participating shelters and the HMIS. The actual subjects are not clients of domestic violence ("DV") homeless shelters, but are clients of homeless family shelters (not domestic violence specific). Using non-DV shelters allowed us to compare computed de-identified results with results derived manually using fully identified data.

A downside to using non-DV shelters is that differences in data collected in DV versus non-DV shelters may exist and would not reflect in results. Therefore, the generalizability of these experiments assume there is no difference between DV and non-DV data collection. This assumption seems reasonable given perceived similarities in client populations. (See Section 10.3.6 for a field-level compliance comparison.)

Below is a description of participants.

- Iowa Institute for Community Alliances, participated in its role as the Planning Office or CoC in Des Moines, Iowa.
- HMIS in DesMoines, Iowa participated as a Shelter in its role to de-duplicate across Shelters and the HMIS. These are the same system administrator at Iowa Institute for Community Alliances.
- House of Mercy in Des Moines, Iowa participated as a Shelter.
- New Directions in Des Moines, Iowa participated as a Shelter.
- YWCA in Des Moines, Iowa participated as a Shelter.

For the remainder of this section, the term "Clients" refers to the Clients represented in data, even though they are not actual DV clients. The term "Shelters" refers to House of Mercy, New Direction, YWCA, and sometimes HMIS. Other times, the HMIS is identified separately. The inclusion or exclusion should be obvious by context. The term "Planning Office" refers to the CoC for DesMoines, Iowa.

10.2.1. Data

Data used in the experiments consisted of retrospective Client data (January through June 2006). Shelters previously provided these records to the HMIS for producing an AHAR. Below is further description of data content and handling.

Shelter	Gold Standard Number of Records	Test Database Number of Records	Modified Test Database Number of Records
HMIS	1937	1937	1937
House of Mercy	59	59	59
New Directions	132	132	132
YWCA	------	------	66
Total	2128	2128	2194

Figure 71. Number of Client records in Gold Standard and Test databases by participant. The Gold Standard Database includes manual corrections and inclusion of missing information for those records known to be of the same Clients. The Test Database lacks these modifications, containing the original errors and omissions. The Modified Test Database is the same as the Test Database with 66 records added to generate more common visits across participants. All databases have the same 1570 distinct Clients.

1 The likelihood that the fields contain omissions or errors is small.

2 The likely number of possible distinct combination of values across the fields must be sufficiently large to be unique for each Client.

3 The Client is likely to provide the same values for the fields at each Shelter.

Figure 72. Conditions for selecting fields for Client source information.

Privacy

In order to produce the initial dataset and to analyze some of the experimental results, personnel needed access to identifiable Client information. The only persons who had such access to identifiable data was the existing HMIS and Shelter personnel from whom the data originated.

Gold Standard Database

System administrators[22] at the HMIS took on the laborious task of extracting identifiable Client data from the HMIS originally contributed by House of Mercy and New Directions. System administrators then manually reviewed the data, manually correcting errors and entering omissions, so that records believed to belong to the same person had accurate information in fields that may form the basis of generating UIDs. The fields subject to correction were *first name*, *last name*, *gender*, and *date of birth*. The total number of records was 2128 for 1570 distinct Clients. This comprised our "Gold Standard" database. The data elements are the Universal Data Elements, including name (as *first name* and *last name* fields) and *Social Security number* (see Figure 5). Figure 71 lists the total records by Shelter.

Test Database

The Test Database contains the same records as the Gold Standard Database, except the records are in their originally unchanged form. None of the values reflect the manual cleaning done in

[22] Eileen Mitchell, HMIS system administrator for the HMIS in Des Moines, Iowa, performed the labor of producing the Gold Standard Database and supervised its use.

the Gold Standard Database. Secondly, the Test Database includes an additional 66 records assigned to the YWCA in order to provide additional visits occurring across more Shelters. The total number of records was 2194 for the same 1570 distinct Clients. The data elements are the Universal Data Elements, including name and Social Security number (see Figure 5). Figure 71 lists the total records by Shelter.

Modified Test Database

The Modified Test Database is a copy of the Test Database with a major change to fields and records. Changes to the fields include dropping, modifying, and re-ordering. Specific fields dropped: *last name* and *Social Security number. Fields changed: first name* to be only the first three letters of the first name. The order of fields is: *first 3 letters of the first name*, *date of birth*, *year of birth*, *race*, *gender*, *veteran*, *disability*, *prior residence type*, *prior residence days*, *ZIP*, *entry date*, *exit date*, *provider ID*, *group ID*, and *program ID*. (See Figure 5 for field descriptions.) The Client source information is the two leftmost fields, *first 3 letters of the first name* and *date of birth*. The remaining fields, *year of birth* through *program ID*, comprise the Universal Data Elements. Records added: 66 records assigned to the YWCA in order to provide additional visits occurring across more Shelters. The total number of records was 2194 for the same 1570 distinct Clients. Figure 71 lists the total records by Shelter.

10.3 Experiments: Client source information

A key component in de-duplicating UIDs is the Client source information used to construct the UIDs. Fields having omissions or errors can render UIDs useless. Experiments in this section compared traditional and proposed choices for constructing UIDs.

Figure 72 lists three conditions for fields to satisfy to be good choices for Client source information.

Problem Statement.

> *Given traditional and proposed ways of constructing UIDs (see Sections 10.3.1, Section 10.3.2, Section 10.3.3, and Section 10.3.4), determine which ways best satisfy the three conditions for constructing UIDs listed in Figure 72.*

The next four subsections describe different ways to construct UIDs.

Position	Content
1	First letter of first name
2	First letter of last name
3	Third letter of last name
4	First letter of gender
5	Date of Birth yyyymmdd (or all 0's if present)
12	Soundex of first name
16	Soundex of last name

Figure 73. Servicepoint Client UID encoding. The result is a 20 character code.

Step	Description
1	Copy the first letter of the string
2	Remove all occurrences of the following unless it is the first letter of the string: a, e, h, i, o, u, w, y
3	From the second letter forward, assign the following number to letters: for b, f, p, v, assign 1 for c, g, j, k, q, s, x, z, assign 2 for d, t, assign 3 for l, assign 4 for m, n, assign 5 for r, assign 6
4	If two or more adjacent numbers repeat, keep only the first.
5	Return the first four characters, padding 0's on the right if needed.

Figure 74. Soundex algorithm. Given a string, the Soundex algorithm provides a 4-character code. Examples: Washington (W252), Robert and Rupert (R163).

Position	Content
1	First letter of first name
2	First letter of last name
3	Third letter of last name
4	Date of Birth yyyymmdd (or all 0's if present)

Figure 75. Servicepoint Client UID encoding variant. This version differs from Figure 73 by not including gender or Soundex. The results is a 11 character code.

Position	Content
1	First three letters of first name
4	Date of Birth yyyymmdd (or all 0's if present)

Figure 76. Privacert proposed Client UID encoding. The result is an 11 character code.

10.3.1. Social Security Number

The Social Security number is perhaps the most common way to reference people in data. This is a 9-digit value, being uniquely assigned to most people in the United States.

10.3.2. Servicepoint Client Unique ID

The most common UID used within HMIS systems, and used by the HMIS in Iowa, is the Servicepoint Client Unique ID.[23] A Servicepoint Client Unique ID is 20 characters, encoded as described in Figure 73.

The Servicepoint encoding uses Soundex[33], which is a phonetic algorithm for encoding names in 4 characters. The Soundex algorithm appears in Figure 74.

10.3.3. Servicepoint Client Unique ID Variant

A simple variant to the Servicepoint UID encoding described above in Section 10.3.2 does not include gender or Soundex. The result is a 11 character encoding, as described in Figure 75.

10.3.4. Proposed Privacert Method

While the PrivaMix Demonstration System works with any Client source information, Privacert proposed one combination of fields for consideration using only the first name and the date of birth. The result is a 11 character encoding, as described in Figure 76.

10.3.5. Experimental design

Using the Test Database, proposed method for constructing UIDs were compared in terms of addressing the three conditions described in Figure 72. Results appear in Section 10.3.6.

[23] Servicepoint is a product of Bowman Systems, servicing more than 30,000 clients in 45 states. They are a national leader in providing HMIS services. For more information, see http://www.bowmansystems.com/products.html.

10.3.6. Results

The following results: (1) compare data compliance of domestic violence shelters to non-DV shelters, (2) report the number of blank values found in the fields of interest to the four methods of constructing UIDs mentioned above; (3) report the number of records effected by data discrepancies in fields relied on by the four methods; and, (4) a summary of how each of the four methods address the three conditions identified as important to the construction of UIDs.

Comparison of DV to non-DV data compliance.
Figure 77 shows previously reported results compiled by Abt[24] from Iowa's Planning Office that compare percentages of missing information in Iowa DV versus non-DV. Data for DV shelters result from on site visits and therefore reflect data maintained by DV shelters internally. Because of the changes to the Universal Data Elements (see Figure 5), first and last name is not required, so DV shelters only collect these fields a half to one-quarter of the time. DV shelters rarely collect Social Security numbers. Values routinely appear for dates of birth. The accuracy of none of the values is known with the ad hoc observation that many dates of birth share the same January 1 value, but with different years.

Number of blank values.
Figure 78 shows the number of blank values found in noted fields in the Test Database, as compiled and previously reported by Abt [34]. Most records lacked a middle initial. Of the fields used by the four methods for constructing UIDs described above, most values were present.

Comparison of UIDs effected by bad or missing data.
Figure 79 shows a comparison of the number of UIDs effected by bad or missing data in the Test Database, as compiled and previously reported by Abt [34]. The comparison is between Servicepoint (Section 10.3.2) and the proposed Privacert (Section 10.3.4) methods. In total, 88 UIDs were negatively impacted using Servicepoint's method compared to only 56 for the proposed Privacert method. Because the proposed Privacert method uses fewer fields that can contain bad or missing data, it performed better. Errors found in the last names or gender fields resulted in bad UIDs for the Servicepoint method while having no adverse effect on the proposed Privacert method.

Summary of UID methods.
Figure 80 compares results of the four methods of UID construction (Section 10.3.1, Section 10.3.2, Section 10.3.3, and Section 10.3.4) in terms of the three conditions found important to constructing UIDs (Figure 72).

Condition 1: The fewer the number of UIDs adversely effected by omission of errors found in the data, the better the method's performance. Values are copied from the earlier results in Figure 78 and Figure 79 for the SSN (264), Servicepoint (88), and proposed Privacert (56) methods. The 84 UIDs negatively effected by omissions or errors using the Servicepoint variant

[24] Results that are noted in this writing as being compiled and previously reported by Abt appear in [34]. During the Iowa experiment, some analyses required access to identifiable HMIS data. This was done by Brian Sokol at Abt under the supervision of the system administrators of Iowa's HMIS. They computed their results independent of this author for privacy reasons and for the benefit of having an independent third party perform such analyses. Results of analyses done by this author are those appearing without credit to Abt.

(Servicepoint2) was inferred from the 88 effected by Servicepoint. The Servicepoint variant does not use gender, which accounted for 4 errors in Figure 79. Overall, the proposed Privacert method performed best.

Condition 2: The larger the number of distinct combinations of the fields, the greater the number of Clients within a CoC that can use the method.

Assuming all possible digits are possible with a SSN gives 10^9 possible values.

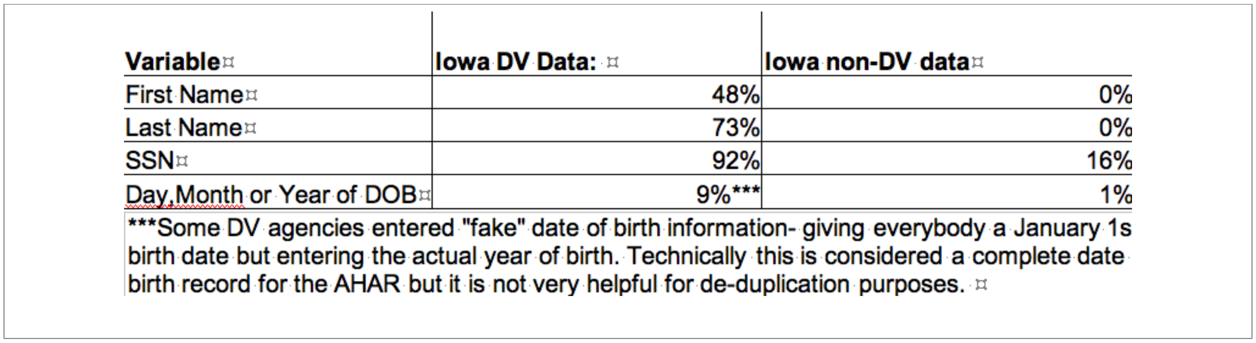
The Servicepoint encoding on has 26 * 2 * 36500 * 156 * 156 = 46 * 10^9 possible values. There are 26 different letters, 2 different genders, 36,500 dates of births (assuming 100 year age range), and 156 possible Soundex values. Using the first letter of the first name and the first letter of the last name is redundant with the Soundex code so those values are not included.

The Servicepoint variant has 26 * 26 * 26 * 36500 = 641 * 10^6 possible values.

Similarly, the proposed Privacert method has 26 * 26 * 26 * 36500 = 641 * 10^6 possible values.

In summary, Social Security numbers and the Servicepoint encoding can accommodate the most number of Clients within a CoC. The Servicepoint variant and the proposed Privacert method are comparable. The numbers computed are the maximum possible. Not all letters are equally likely in names and not all dates of birth over a 100 year range are equally likely to be Clients. Nonetheless, all four methods seem reasonable for the subjects in this study.

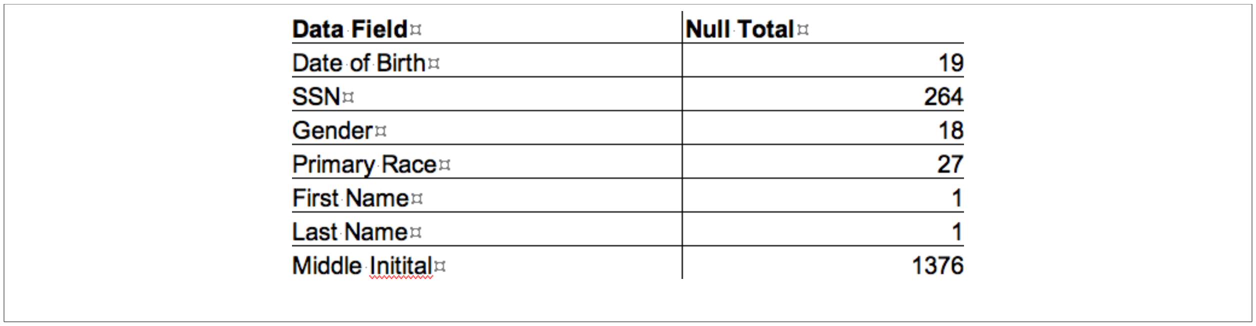
Condition 3: Consistency of values cannot be measured without de-duplication which was not done in this set of experiments (see Section 10.4).



Variable	Iowa DV Data:	Iowa non-DV data
First Name	48%	0%
Last Name	73%	0%
SSN	92%	16%
Day,Month or Year of DOB	9%***	1%

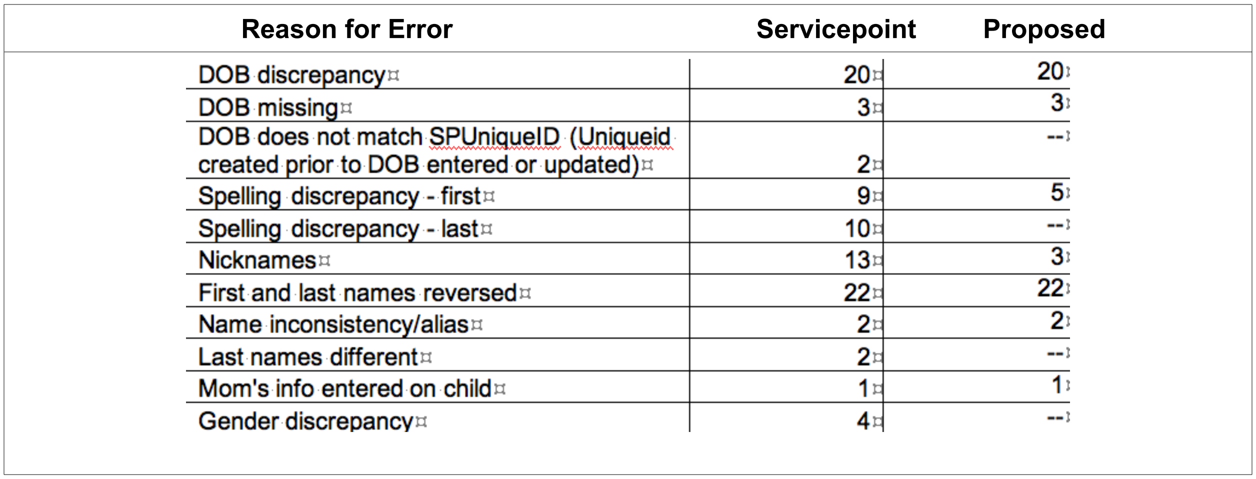
***Some DV agencies entered "fake" date of birth information- giving everybody a January 1s birth date but entering the actual year of birth. Technically this is considered a complete date birth record for the AHAR but it is not very helpful for de-duplication purposes.

Figure 77. Percent of missing data for DV and non-DV shelters. Fields are first name, last name, Social Security numbers (SSN), and dates of birth (DOB) of Clients. Courtesy Abt Associates [34].



Data Field	Null Total
Date of Birth	19
SSN	264
Gender	18
Primary Race	27
First Name	1
Last Name	1
Middle Initital	1376

Figure 78. Number of missing values found in Test Database. A missing or "null" value has no value appearing in the database. Counts based on 2128 records. Courtesy Abt Associates [34].



Reason for Error	Servicepoint	Proposed
DOB discrepancy	20	20
DOB missing	3	3
DOB does not match SPUniqueID (Uniqueid created prior to DOB entered or updated)	2	--
Spelling discrepancy - first	9	5
Spelling discrepancy - last	10	--
Nicknames	13	3
First and last names reversed	22	22
Name inconsistency/alias	2	2
Last names different	2	--
Mom's info entered on child	1	1
Gender discrepancy	4	--

Figure 79. Comparison of UIDs effected by bad or missing data. Compares Servicepoint's UID construction (Section 10.3.2) to the proposed Privacert method (Section 10.3.4) using records in the Test Database. Counts based on 2128 records. Courtesy Abt Associates [34].

Method	Omissions or Errors in Fields	Number of distinct combinations	Consistency of values
SSN	264	10^9	84.60%
Servicepoint	88	$46 * 10^9$	96.40%
Servicepoint2	84	$641 * 10^6$	97.50%
Proposed	56	$641 * 10^6$	97.90%

Figure 80. Comparative summary of four UID methods in real-world data. Results are using only Social Security numbers (SSN), the Servicepoint method (Section 10.3.2), the Servicepoint variant (Section 10.3.3), and the proposed Privacert method (Section 10.3.4) in the Test Database (Section 10.2). The fewer the number of UIDs adversely effected by omission of errors found in the data, the better the method's performance. The larger the number of distinct combinations of the fields, the greater the number of Clients within a CoC that can use the method. Consistency of values is 1-error percentage in Figure 83 (Section 10.4.2). Counts based on 2128 records.

10.4 Experiments: de-duplication

A primary motivation for this work is the utility of de-duplicating UIDs in order to match Client visit information across Shelters. Experiments in this section measured the performance of the PrivaMix Demonstration at de-duplicating real-world data.

10.4.1. Experimental design

Records in the Modified Test Database were divided into smaller databases, one for each of the participants that originally provided the information. The result was four smaller databases, one for each of the three Shelters, and one for the HMIS. Figure 71 shows the distributions of records.

There was a total of 2194 records, with more than half (1937) originating from the HMIS alone. Records originating in the HMIS in this experiment reflect "non-DV" services provided to DV and non-DV clients, indistinguishably. Client information held in the Shelter databases represent "DV" Clients in this experiment. The goal is to de-duplicate visits across DV and non-DV services.

Problem Statement.

> *Given three Shelters, an HMIS, and a Planning Office participating in a PrivaMix network, use the PrivaMix Demonstration System to de-duplicate visits.*

Each of the smaller databases was loaded onto a laptop as a comma-delimited file. Figure 68 lists the fields that comprised the comma-delimited file. The first two fields denote the Client source information. These are *FirstName* and *DateOfBirth*. The remaining 13 fields of the Universal Data Elements (Figure 5) stored values describing the service received by the Client.[25]

The HMIS used the faster Dell machine. The remaining Shelters and the Planning Office used the Toshiba machines. The Dell also used the Sprint wireless modem card, whereas the other Shelters and the Planning Office used the Verizon cards.

The files were saved on each computer with a filename matching the default setting in the PrivaMix Demonstration System. The number of leftmost fields designated to use as Client source information for generating UIDs (2) also matched the default setting in the PrivaMix Demonstration System. The goal was not to assess the flexibility of the software or user's ability to use the laptop per se. Operation was made to be as simple as possible. Upon powering on the machine, the broadband wireless card automatically connected to the Internet and the PrivaMix software loaded. The user need only power on the machine and click the De-duplicate button at the designated time. See Quick Start in Appendix A (page 5 of 16).

The three Shelter machines and the HMIS machine contained the Shelter Edition of the PrivaMix Demonstration System (Section 9). The Planning Office machine contained the CoC Edition of the PrivaMix Demonstration System (Section 9).

Personnel from the HMIS physically visited each Shelter, one at a time. The machine containing that Shelter's information was left with the Shelter. A five minute discussion reviewed the

[25] Personnel from the HMIS actually loaded the data onto the laptops and maintained control of the laptops until providing the machines to the respective Shelters.

security of the machine, the agreed upon time at which de-duplication would occur, the process of powering on the machine, and the need to click the "De-duplicate" button to start.

The agreed upon date and time to start the process was June 5, 2007 at 3pm. At that time, each participant would power on their respective machine at their physical location and then start the de-duplication process.

Once the process begins, there are four distinct phases.

In Phase I, all participants, including the Planning Office, start their machine and click the De-duplicate button. The software will register the machine by sharing IP addresses among only those computers previously known to be participants in the PrivaMix network. See Section 9.1.1 for details.

After all machines complete Phase I, the machines automatically begin Phase II. Each of the Shelter and HMIS machines load the comma-delimited file containing Client information specific to that Shelter or HMIS. The machine then randomly selects a private value (see Section 9.3). The machine then computes UIDs and forwards results to the Planning Office machine. See Figure 68 for examples.

Once the Planning Office machine receives the Client information from the other machines, it initiates mixing, which constitutes Phase III. The Planning Office contacts each Shelter and HMIS, one a time, to mix UIDs and mixes from the other Shelters and HMIS. See Section 9.7 for details.

Once all Shelters and HMIS have mixed all UIDs, Phase IV, the last phase begins. The Planning Office machine de-duplicates UIDs by matching records based on complete mixes. It then re-numbers UIDs and GroupID values sequentially. Finally, comma-delimited results are then stored to the hard drive of the Planning Office machine. See Figure 69 for examples. See Section 9.8 for processing details.

10.4.2. Results

Results: (1) show the time taken for each phase of de-duplication; and, (2) compare de-duplication results.

Time spent.
From the start of the de-duplication process at the designated time until the delivery of the de-duplicated results on the Planning Office machine took 71 minutes. During this time, Shelters forwarded 2253 Client records, thereby mixing 2253 UIDs over four Shelters and HMIS machines. Figure 81 shows the amount of time spent in each phase, as compiled and previously reported by Abt [34].

Despite the Toshiba computers being identical and having to mix the same number of UIDs, the first Shelter took twice as long (20 minutes) to complete mixing in comparison to the other two Shelters using Toshiba machines (11 minutes). The reason for the discrepancy is not clear. It may delay in the Internet connection or start-up overhead. The Dell laptop, used by the HMIS, is much faster. It took only 3 minutes!

During operation, one of the Shelters (House of Mercy) accidentally shut down their machine and had to restart. The program had not anticipated a restart in Phase 1. The result was the double inclusion of their 59 records. It was as if each of their Clients visited them twice.

Phase	Description	Time Completed
Phase 1	All participants run PrivaMix software.	03:00:00 PM
	Network registration.	03:01:00 PM
	Restart by House of Mercy after accidental shutdown.	03:08:00 PM
Phase II	Compute UIDs and forward data	03:25:00 PM
Phase III	Mix Shelter 1 (House of Mercy)	03:45:00 PM
	Mix Shelter 2 (HMIS)	03:48:00 PM
	Mix Shelter 3 (New Directions)	03:59:00 PM
	Mix Shelter 4 (YWCA)	04:10:00 PM
Phase IV	Produce de-duplicated result (CoC)	04:11:00 PM

Figure 81. PrivaMix Demonstration: time spent per phase. Courtesy Abt Associates [34].

Manual de-duplication of Gold Standard Database.

Figure 82 reports manually produced de-duplication results on the Gold Standard Database for three different UID methods, as compiled and previously reported by Abt [34]. Manual de-duplication was done by constructing UIDs with a noted method and then matching results to get a distinct count. Servicepoint and the proposed Privacert method both provided an accurate de-duplicated count of 1570 Clients. Matching Social Security numbers (SSNs) only found 1330 of the Clients because 240 records had no SSN.

Manual de-duplication of Test Database.

Figure 83 reports manually produced de-duplication results on the Test Database for four different UID methods, as compiled and previously reported by Abt [34]. Manual de-duplication was done by constructing UIDs with a noted method and then matching results to get a distinct count. Some Social Security numbers (SSN) were missing, leading to an undercount by that method. All other methods had an over count. The proposed Privacert method performed best, though comparable to the Servicepoint variant. Because Privacert did not use the last name field, an error found there did not effect its performance as it did with the Servicepoint variant.

Consistency of values.

The third condition for selecting fields for Client source information (Figure 72) involves computing the likelihood a Client will provide the same values at each Shelter visited, based on the fields used by the noted UID method. This writing terms this “the consistency measure.” The error percentage in Figure 83 provides a basis for a consistency measure as the inverse of the error percentage, computed as [1.0 – (error percentage)]. This measures the accuracy of de-duplication across records from different Shelters. In manual de-duplication of records in the Test Database, Social Security numbers had the worst consistency (84.6%) because the value was sometimes missing. The proposed Privacert method had the best consistency (97.9%). The Servicepoint variant was comparable (97.5%). The Servicepoint method did a little worse (96.4%). These values appear in Figure 80, as part of a comparative summary of UID methods in terms of selecting Client source information.

PrivaMix de-duplication of Modified Test Database.

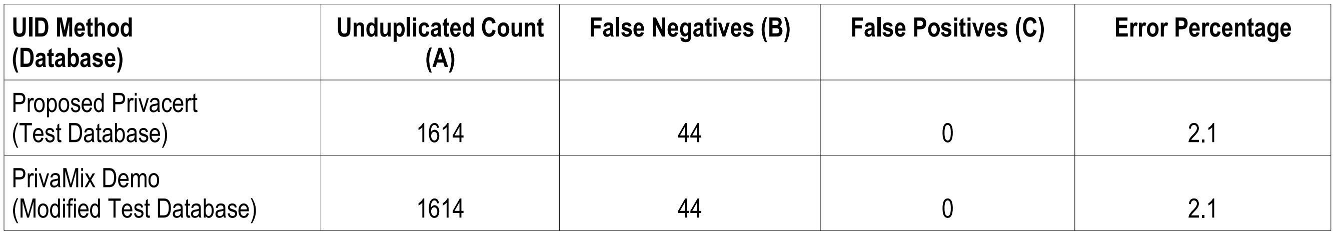
Figure 84 compares de-duplication results from the PrivaMix Demonstration System on the Modified Test Database with the earlier manual results on the Test Database, as compiled and previously reported by Abt [34]. The PrivaMix Demonstration System performed exactly the same as the manual results predicted (Figure 83). The System did not introduce any errors and made the same decisions on all records as constructing encodings in plain text (Figure 76). Shelters constructing UIDs from the plain text did not generate any mismatches. Mixing UIDs and then matching on the complete mixes introduced no omissions or errors. The PrivaMix Demonstration System performed exactly as if plain text encoding was used even though the Client information was provably never shared with the Planning Office or the other Shelters.

UID Method	Unduplicated Count (A)	False Negatives (B)	False Positives (C)	Error Percentage
SSN	1330	0	240	11.3
Servicepoint	1570	0	0	0
Proposed Privacert	1570	0	0	0

Figure 82. Manual de-duplication results on Gold Standard Database. In the 2128 records, the number of distinct Clients is 1570. Some Social Security numbers (SSN) were missing, leading to an undercount by that method. A false positive results when two distinct Clients are counted as one. A false negative results when a record belonging to a known Client is missed. The error percentage is (B) + (C)/2128 * 100. Courtesy Abt Associates [34].

UID Method	Unduplicated Count (A)	False Negatives (B)	False Positives (C)	Error Percentage
SSN	1360	59	269	15.4
Servicepoint	1646	76	0	3.6
Servicepoint 2	1619	51	2	2.5
Proposed Privacert	1614	44	0	2.1

Figure 83. Manual de-duplication results on Test Database. In the 2128 records, the number of distinct Clients is 1570. A false positive results when two distinct Clients are counted as one. A false negative results when a record belonging to a known Client is missed. The error percentage is (B) + (C)/2128 * 100. Courtesy Abt Associates [34].



UID Method (Database)	Unduplicated Count (A)	False Negatives (B)	False Positives (C)	Error Percentage
Proposed Privacert (Test Database)	1614	44	0	2.1
PrivaMix Demo (Modified Test Database)	1614	44	0	2.1

Figure 84. PrivaMix de-duplication results. The predicted results (top row) match the actual results (bottom row) exactly. A false positive results when two distinct Clients are counted as one. A false negative results when a record belonging to a known Client is missed. The error percentage is (B) + (C)/2128 * 100. Courtesy Abt Associates [34].

10.5 Summary

In a real-time experiment with three shelters, an HMIS and a Planning Office, a “PrivaMix Demonstration System” computed an accurate unduplicated accounting using real-world data from homeless programs in Des Moines, Iowa (“the Iowa Experiment”). Here is a summary of experimental results.

The experiment used laptops with wireless broadband network, with the software loaded and pre-configured for operation. Standardizing the machines allowed the experiments to focus efficiently and narrowly on performance.

Subjects were clients whose data appeared a participating shelters and the HMIS in a previous six-month time period. The actual subjects are not clients of domestic violence (“DV”) homeless shelters, but are clients of homeless family shelters (not domestic violence specific). Using non-DV shelters allowed us to compare computed de-identified results with results derived manually using fully identified data. Of course, the generalizability of these experiments assume there is no difference between DV and non-DV data collection.

A key component in de-duplicating UIDs is the Client source information used to construct the UIDs. Fields having omissions or errors can render UIDs useless. While the PrivaMix Demonstration System works with any Client source information, Privacert proposed to use the first three letters of the first name and the date of birth. Experiments compared Privacert's proposed method with using Social Security numbers, and two methods currently in use by Servicepoint. Privacert's method encountered fewer fields having omissions or errors than the other methods, and used fields in which clients provided more consistent values than the fields used by the other methods. In performing an unduplicated accounting, the Privacert method had the lowest number of errors.

After constructing UIDs, shelters, the HMIS, and Planning Office conducted a real-time duplication using the laptops located at their facilities. The PrivaMix Demonstration System performed exactly as if plain text was used even though sensitive Client source information was provably never shared with the Planning Office or the other Shelters. No errors were introduced.

11. Identifiability of Iowa's De-duplicated Results

The goal of this work is to accomplish de-duplication with guarantees of privacy protection. The PrivaMix Demonstration System accurately de-duplicates (Section 10), and provably provides privacy protection of the UID, throughout the UID construction and de-duplication processes (see Section 9). However, data linkage problems may still exist (Section 4.2) because they do not involve the UIDs, but the data elements that are shared. While the UIDs have provable protection, the shared data elements may be vulnerable. This section examines the uniqueness and re-identification risks associated with shared data elements.

Problem Statement.

> *Given de-duplicated results, compute the uniqueness of Clients and describe possible re-identification strategies.*

11.1 Statistical description of Iowa's demographic elements

This section reports distributions of Client demographics found in the Iowa data that was de-duplicated by PrivaMix.

11.1.1. Analysis design

De-duplicated Demographics Database.

This section reports distributions of Client demographics found in the Iowa data that was de-duplicated by PrivaMix. This writing terms this data the "De-duplicated Demographics Database." Figure 85 reviews the variations of databases used in the Iowa experiments. The Modified Test Database was the source of PrivaMix de-duplication. The de-duplicated results provided data having the same rows of information as found in the Modified Test Database (Figure 84). The fields are different because Client source fields did not appear in de-duplicated results. Of the fields that do appear, some of them contain demographic values –specifically, *year of birth*, *gender*, *ZIP*, *race*, and *ethnicity*. These are the only fields in the De-duplicated Demographic Database. The rows are the values appearing for the 1614 distinct Clients. In summary, the De-duplicated Demographic Database had 1614 records and 5 fields, where each record represents the demographics of a distinct de-duplicated Client.

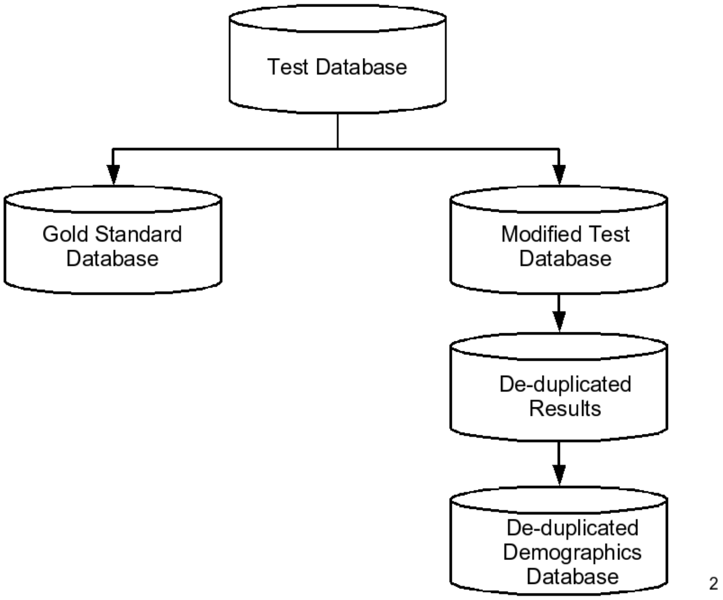


Figure 85. Relationships of databases used in Iowa Experiment. The original Test Database has 2128 records. The Gold Standard Database includes manual corrections to identify 1570 distinct Clients. The Modified Test Database has 66 records added to generate more common visits across participants. After PrivaMix de-duplication, the De-duplicated Results contains a copy of the Modified Test Database with Client source information replaced with sequentially assigned numbers that repeat to identify records belonging to the same Client. The De-duplicated Demographics Database contains only a distinct copy of Client demographic fields in the De-duplicated Results. See Figure 71 for record counts.

11.1.2. Results

Figure 86 displays the ZIP code distribution of the 1614 distinct Clients in the De-duplicated Demographics Database. Not listed are 492 Clients for which no ZIP was found. ZIP 50309 had the most Clients (224). Average number of Clients per ZIP was 7, with a standard deviation of 25. A total of 101 Clients (or 6%) have a unique 5-digit ZIP code.

Figure 87 displays the distribution of the years of births of the 1614 distinct Clients in the De-duplicated Demographics Database. No year of birth was reported for 12 Clients. The most popular year was 1957 (48 Clients). Average number of Clients per year was 21, with a standard deviation of 13. Years of birth from 1901 through 1937 (6 values) are unique.

Figure 88 displays the distribution of gender, race, ethnicity, and race and ethnicity combined for the 1614 distinct Clients in the De-duplicated Demographics Database.

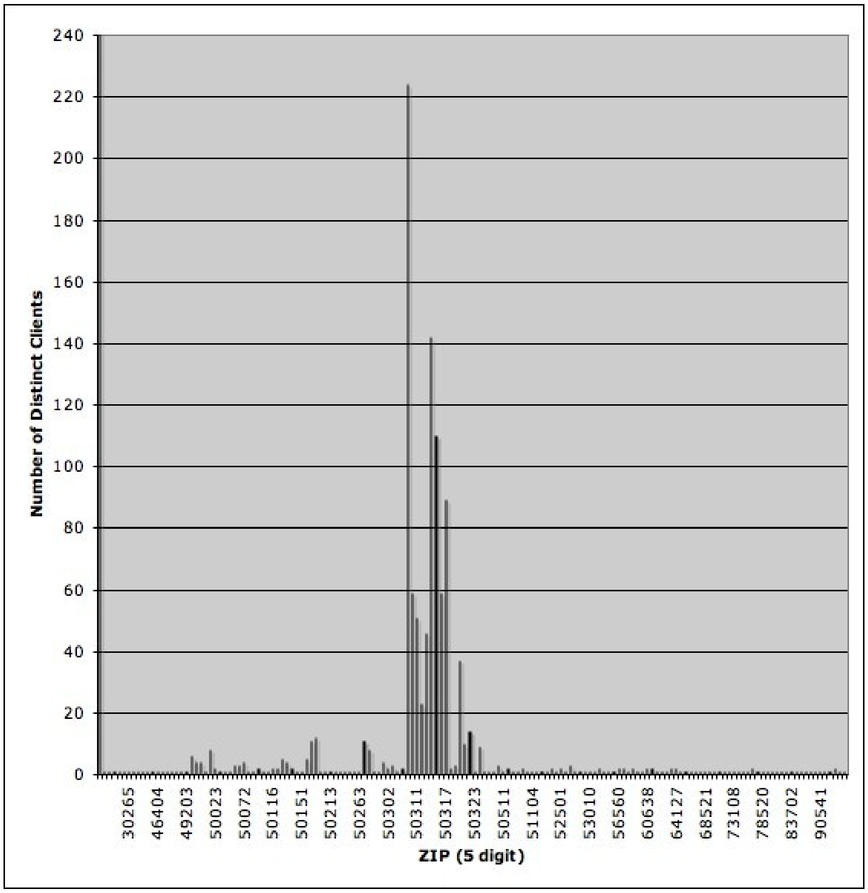


Figure 86. Distribution of 5-digit ZIP codes in Iowa de-duplicated results. Counts based on 1614 de-duplicated clients in the De-duplicated Demographics Database. Not listed are 492 Clients for which no ZIP was found. ZIP 50309 had the most Clients (224). Average number of Clients per ZIP was 7, with a standard deviation of 25.

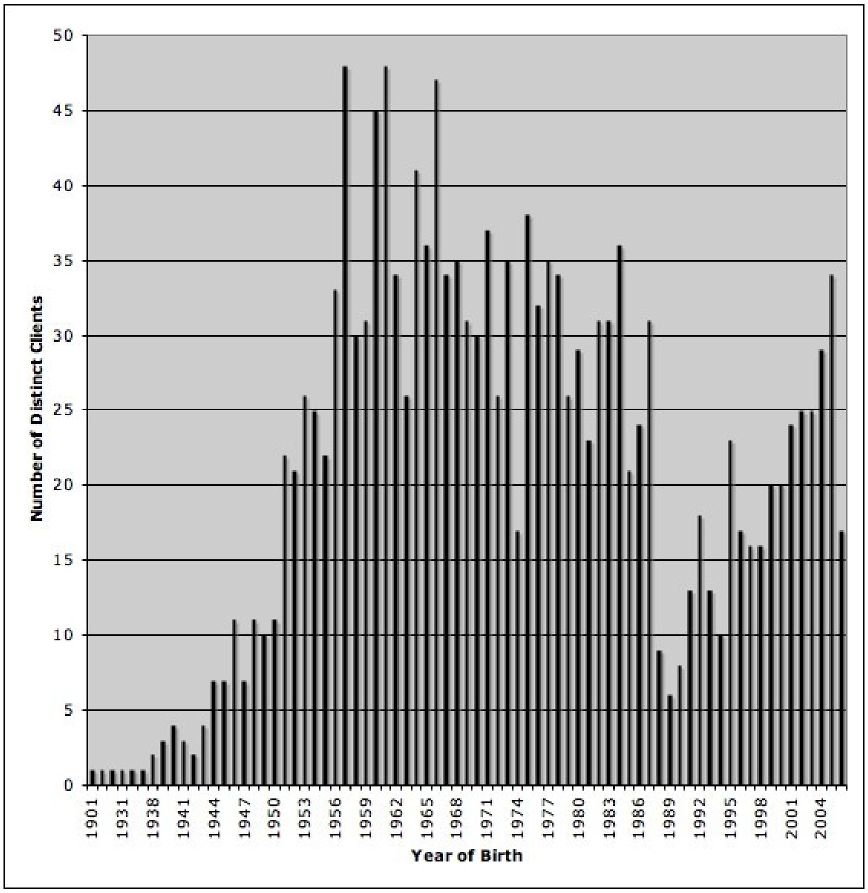
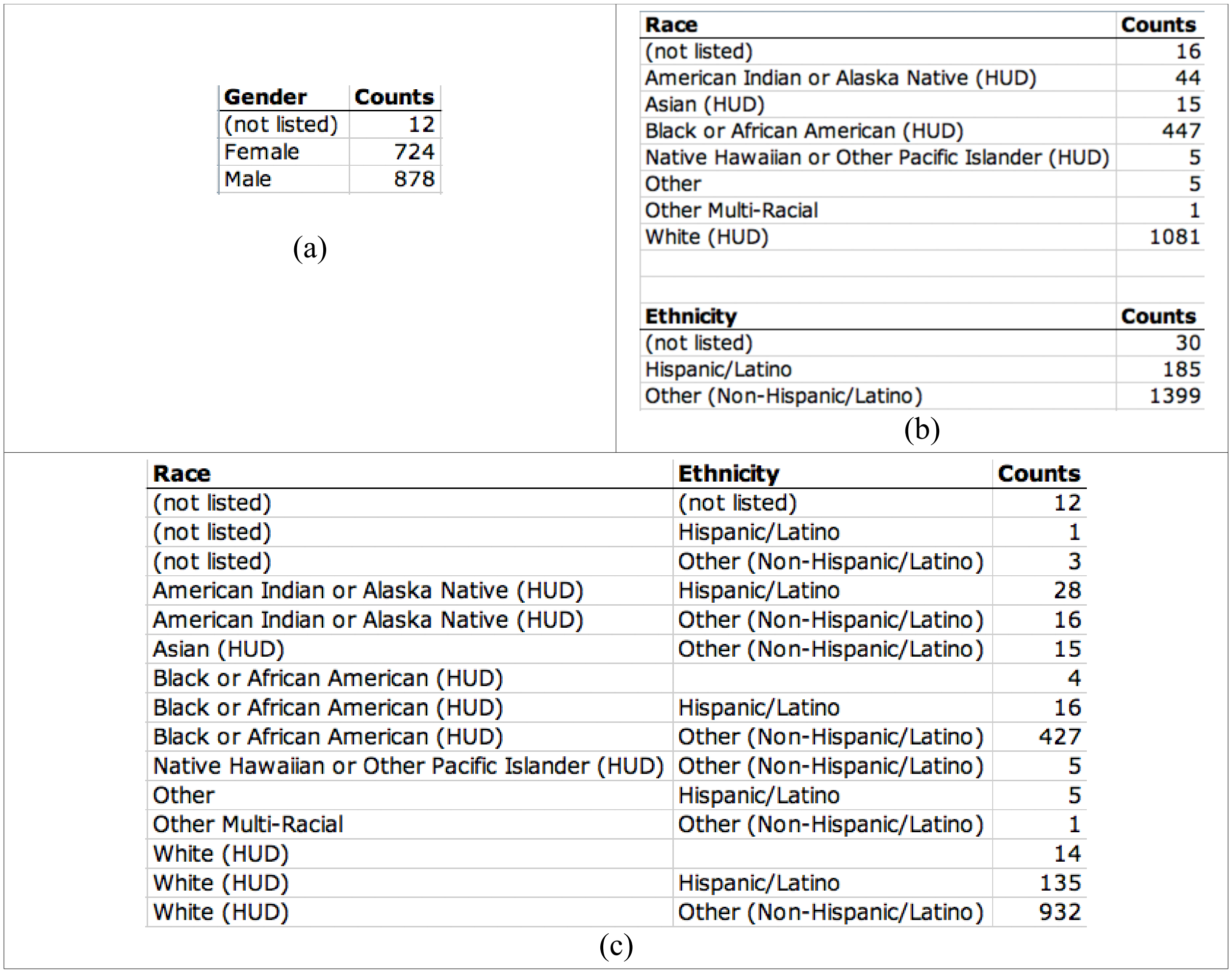


Figure 87. Distribution of Client years of birth in Iowa de-duplicated results. Counts based on 1614 de-duplicated clients in the De-duplicated Demographics Database. No year of birth was reported for 12 Clients. The most popular year was 1957 (48 Clients). Average number of Clients per year was 21, with a standard deviation of 13.



Gender	Counts
(not listed)	12
Female	724
Male	878

(a)

Race	Counts
(not listed)	16
American Indian or Alaska Native (HUD)	44
Asian (HUD)	15
Black or African American (HUD)	447
Native Hawaiian or Other Pacific Islander (HUD)	5
Other	5
Other Multi-Racial	1
White (HUD)	1081

Ethnicity	Counts
(not listed)	30
Hispanic/Latino	185
Other (Non-Hispanic/Latino)	1399

(b)

Race	Ethnicity	Counts
(not listed)	(not listed)	12
(not listed)	Hispanic/Latino	1
(not listed)	Other (Non-Hispanic/Latino)	3
American Indian or Alaska Native (HUD)	Hispanic/Latino	28
American Indian or Alaska Native (HUD)	Other (Non-Hispanic/Latino)	16
Asian (HUD)	Other (Non-Hispanic/Latino)	15
Black or African American (HUD)		4
Black or African American (HUD)	Hispanic/Latino	16
Black or African American (HUD)	Other (Non-Hispanic/Latino)	427
Native Hawaiian or Other Pacific Islander (HUD)	Other (Non-Hispanic/Latino)	5
Other	Hispanic/Latino	5
Other Multi-Racial	Other (Non-Hispanic/Latino)	1
White (HUD)		14
White (HUD)	Hispanic/Latino	135
White (HUD)	Other (Non-Hispanic/Latino)	932

(c)

Figure 88. Distribution of gender, race, and ethnicity in Iowa de-duplicated results. Counts based on 1614 clients in the De-duplicated Demographics Database. Race and Ethnicity values are reported separately in (b) and in combination in (c).

11.2 Uniqueness of demographic combinations in Iowa results

Examining demographics individually, as was done in above in Section 11.1, revealed that some values for year of birth (6) and ZIP (101) appeared only once. These values are unique and therefore the demographics of their 107 Clients are unique. However, demographic fields often combine to jointly yield a greater number of unique combinations. This section reports on the uniqueness of combinations of demographic values occurring in the De-duplicated Demographic Database.

11.2.1. Analysis design

When discussing various ways of counting unique combinations of values, the term "binsize" is useful.

A binsize refers to the number of people to whom a record could ambiguously relate. In the De-duplicated Demographic Database, size people have a distinct year of birth. So, each of these

Clients has a bin size of 1 with respect to year of birth. Similarly, two people were born in 1938 and two people were born in 1942. Each of these Clients has a bin size of 2 with respect to year of birth.

Examining how the number of unique combinations changes as the information becomes less specific is useful in understanding how data elements can have fewer unique combinations. Results in this section will look at the following aggregations of fields:

ZIP	ZIP5	5-digit postal code provided in data
	ZIP4	First 4 digits of postal code (larger geography)
	ZIP3	First 3 digits of postal code (largest geography examined)
Year of Birth	Year of birth	4 digit year of birth provided in data
	Age	Computed as a 2-year range computed using year of birth
	5-year age range	5-year range computed using year of birth
	AHAR age ranges	Age ranges used in AHAR, computed using year of birth. Ranges are: under 1, 1 through 5, 6 through 12, 13 through 17, 18 through 30, 31 through 50,, 51 through 61, and 62 and over.

11.2.2. Uniqueness results

Figure 89 shows the percentage and number of unique combinations of values for various aggregations of ZIP and age. All combinations include gender. Unique combinations of {year of birth, gender, ZIP5} occurred in 36% of Clients (or 580 Clients). As fewer rightmost digits appear in the ZIP, the number of unique combinations decreases. Unique combinations of {year of birth, gender, ZIP4} occurred in 21% of Clients (or 346 Clients). Unique combinations of {year of birth, gender, ZIP3} occurred in 16% of Clients (or 255 Clients).

Similarly, using less specific age information reduces the number of unique combinations. Unique combinations of {age, gender, ZIP5} occurred in 26% of Clients (or 423 Clients). Unique combinations of {age, gender, ZIP5} occurred in 18% of Clients (or 294 Clients). And, unique combinations of {AHAR age ranges, gender, ZIP5} occurred in 18% of Clients (or 294 Clients).

The most aggregated combination of {AHAR age ranges, gender, ZIP3} provided the fewest number of unique combinations, 6% of Clients (or 100 Clients). All combinations of aggregations of ZIP, gender, and age examined revealed unique combinations.

Figure 90, Figure 91, Figure 92, and Figure 93 show the cumulative percentage of population as the binsize increases. The leftmost value in each curve is binsize 1. These are the number of unique occurrences described previously in Figure 89. The rightmost point on each curve occurs when the entire population is included.

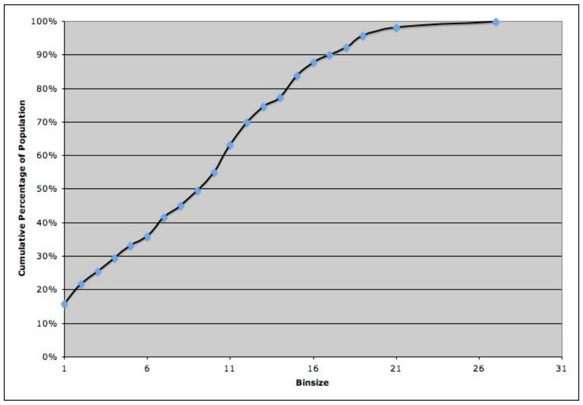
Figure 94 examines combinations that include race and ethnicity. The percentage and number of unique combinations of values for various aggregations of ZIP and age are copied from Figure 89 for comparison. In each case, additionally including race and ethnicity increased the number of unique combinations. A general observation is that including race and ethnicity almost doubles the number of unique combinations.

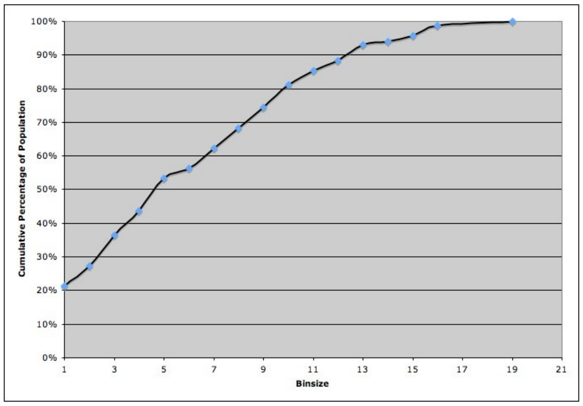
Figure 95, Figure 96, Figure 97, and Figure 98 show the cumulative percentage of population as the binsize increases for combinations that include race and ethnicity. Curves are copied from Figure 90, Figure 91, Figure 92 and Figure 93 for comparison to the curves related to combinations that do not include race and ethnicity.

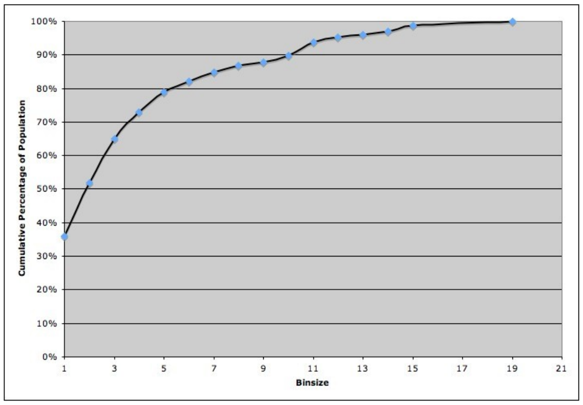
Figure 99, Figure 100, Figure 101, and Figure 102 show the distributions of binsizes for various combinations of demographic values, with and without race and ethnicity.

ZIP3	16% (255)	12% (193)	9% (139)	6% (100)
ZIP4	21% (346)	17% (267)	12% (201)	8% (136)
ZIP5	36% (580)	26% (423)	18% (294)	12% (196)
Gender	**Year of Birth**	**Age**	**5 year ranges**	**AHAR ranges**

Figure 89. Percentage of unique occurrences in combined ZIP, gender, age aggregations in Iowa de-duplicated results. Counts based on 1614 clients in the De-duplicated Demographics Database. Categories of ZIP are 5-digits (ZIP5), the first 4 digits (ZIP4) and the first 3 digits (ZIP3). AHAR age ranges: Under 1, 1 to 5, 6 to 12, 13 to 17, 18 to 30, 31 to 50, 51 to 61, and 62 and above. Age computed as a 2-year range using year of birth. Number of Clients having unique combinations of noted field combination appear in parentheses.

<table>
<tr>
<td><strong>ZIP3</strong></td>
<td>


16% unique (255 Clients)
</td>
</tr>
<tr>
<td><strong>ZIP4</strong></td>
<td>


21% unique (346 Clients)
</td>
</tr>
<tr>
<td><strong>ZIP5</strong></td>
<td>


36% unique (580 Clients)
</td>
</tr>
<tr>
<td><strong>Gender</strong></td>
<td><strong>Year of Birth</strong></td>
</tr>
</table>

Figure 90. Binsize distributions for gender, year of birth and ZIP in Iowa de-duplicated results. Population: De-duplicated Demographics Data (1614 total). ZIP: 5-digits (ZIP5), first 4 digits (ZIP4), first 3 digits (ZIP3).

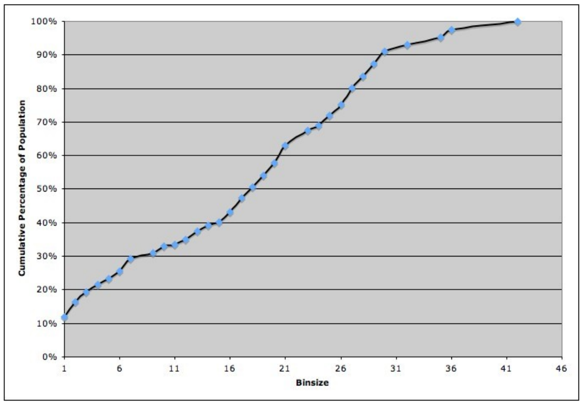
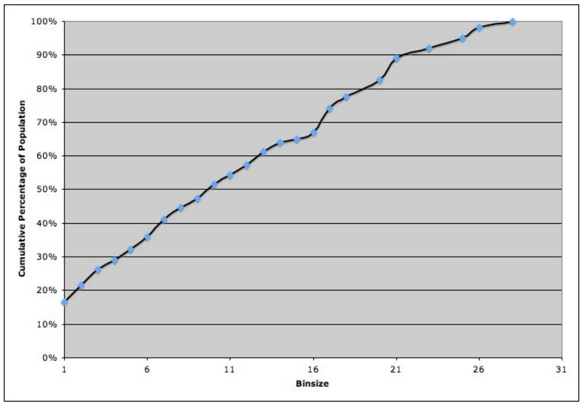
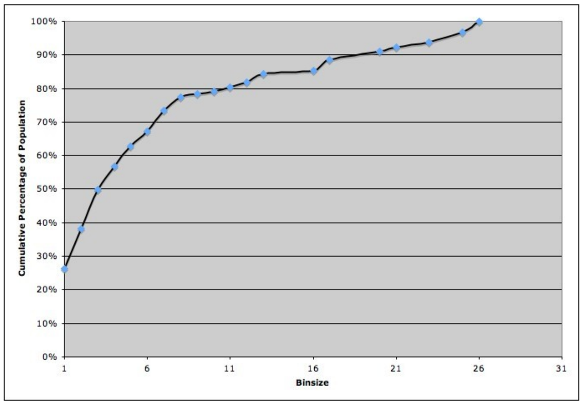
ZIP3	 12% unique (193 Clients)
ZIP4	 17% unique (267 Clients)
ZIP5	 26% unique (423 Clients)
Gender	**Age**

Figure 91. Binsize distributions for gender, age and ZIP aggregations in Iowa de-duplications. De-duplicated Demographics Database (1614 total). 5-digit (ZIP5), first 4 digits (ZIP4), first 3 digits (ZIP3). Age is 2-year range.

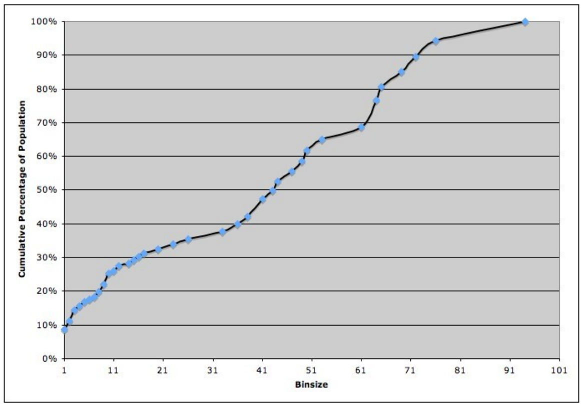
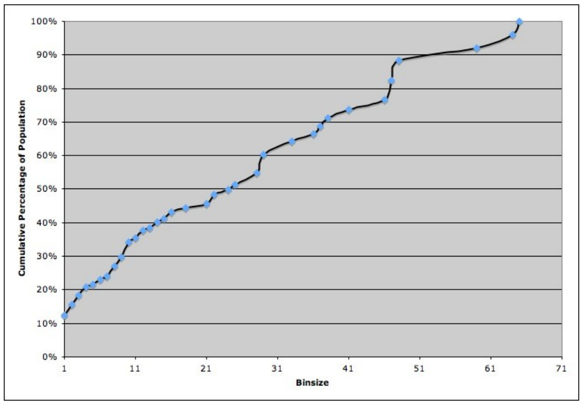
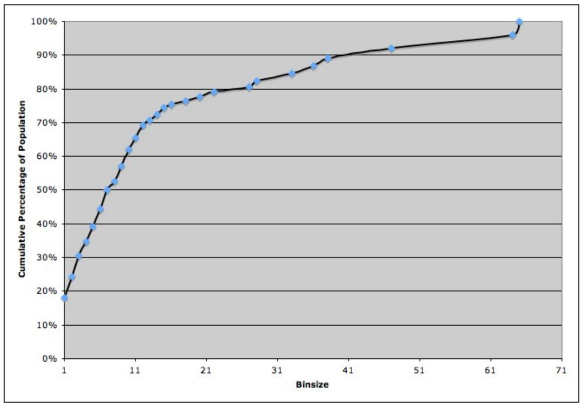
ZIP3	 9% unique (139 Clients)
ZIP4	 12% unique (201 Clients)
ZIP5	 18% unique (294 Clients)
Gender	**5-year Age Ranges**

Figure 92. Binsize distributions for gender, 5-year age ranges and ZIP aggregations in Iowa de-duplications. De-duplicated Demographics Database (1614 total). ZIP: 5-digits (ZIP5), first 4 digits (ZIP4), first 3 digits (ZIP3).

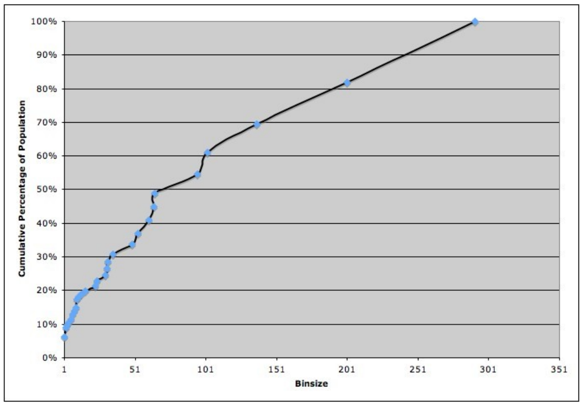
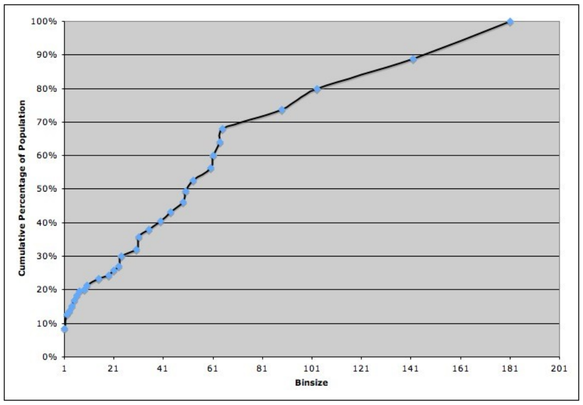
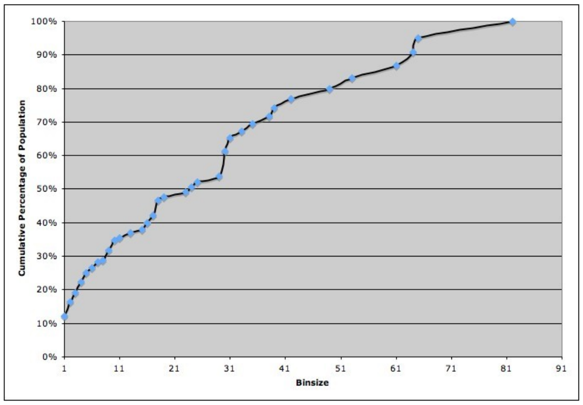
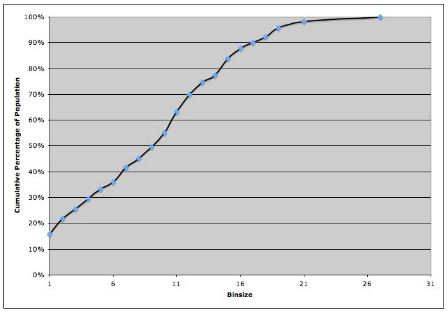
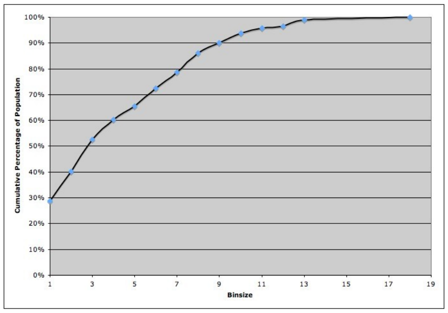
ZIP3	 6% unique (100 Clients)
ZIP4	 8% unique (136 Clients)
ZIP5	 12% unique (196 Clients)
Gender	**AHAR age ranges**

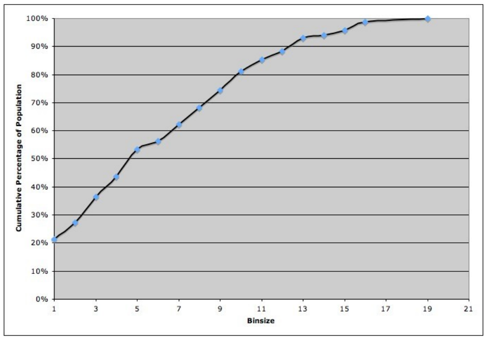
Figure 93. Binsize distributions for combined gender, AHAR age ranges and ZIP aggregations in Iowa de-duplicated results. Counts based on 1614 clients in the De-duplicated Demographics Database. Categories of ZIP are 5-digits (ZIP5), the first 4 digits (ZIP4) and the first 3 digits (ZIP3). AHAR age ranges: Under 1, 1 to 5, 6 to 12, 13 to 17, 18 to 30, 31 to 50, 51 to 61, and 62 and above.

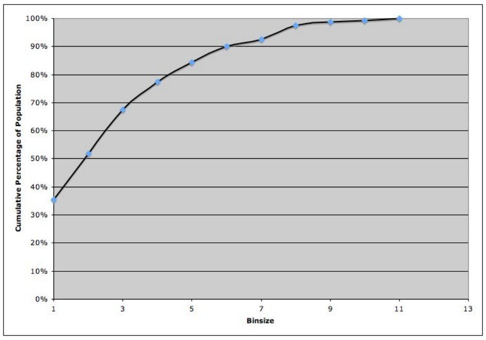
ZIP3	**Gender**	16% (255)	12% (193)	9% (139)	6% (100)
	Gender, Race, Ethnicity	29% (464)	21% (346)	15% (245)	11% (178)
ZIP4	**Gender**	21% (346)	17% (267)	12% (201)	8% (136)
	Gender, Race, Ethnicity	35% (571)	28% (455)	20% (326)	14% (223)
ZIP5	**Gender**	36% (580)	26% (423)	18% (294)	12% (196)
	Gender, Race, Ethnicity	55% (882)	44% (704)	30% (488)	20% (317)
		Year of Birth	**Age**	**5 year ranges**	**AHAR ranges**

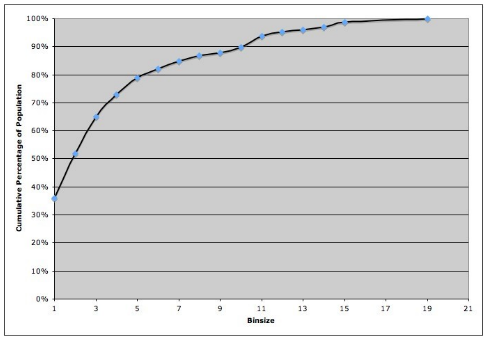
Figure 94. Percentage of unique occurrences in combined demographic aggregations in Iowa de-duplicated results. Counts based on 1614 clients in the De-duplicated Demographics Database. Categories of ZIP are 5-digits (ZIP5), the first 4 digits (ZIP4) and the first 3 digits (ZIP3). AHAR age ranges: Under 1, 1 to 5, 6 to 12, 13 to 17, 18 to 30, 31 to 50, 51 to 61, and 62 and above. Age computed as a 2-year range using year of birth. Number of Clients having unique combinations of noted field combination appear in parentheses.

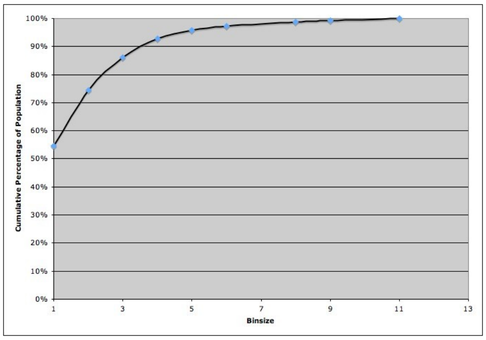
<table>
<tr>
<td><b>ZIP3</b></td>
<td>

16% unique (255 Clients)</td>
<td>

29% (464 Clients)</td>
</tr>
<tr>
<td><b>ZIP4</b></td>
<td>

21% unique (346 Clients)</td>
<td>

35% (571 Clients)</td>
</tr>
<tr>
<td><b>ZIP5</b></td>
<td>

36% unique (580 Clients)</td>
<td>

55% unique (882 Clients)</td>
</tr>
<tr>
<td rowspan="2"></td>
<td><b>Gender</b></td>
<td><b>Gender, Race, Ethnicity</b></td>
</tr>
<tr>
<td colspan="2"><b>Year of Birth</b></td>
</tr>
</table>

Figure 95. Comparison of cumulative binsize distributions for combined demographics, year of birth and ZIP aggregations in Iowa de-duplicated results. Counts based on 1614 clients in the De-duplicated Demographics Database. Categories of ZIP are 5-digits (ZIP5), the first 4 digits (ZIP4) and the first 3 digits (ZIP3).

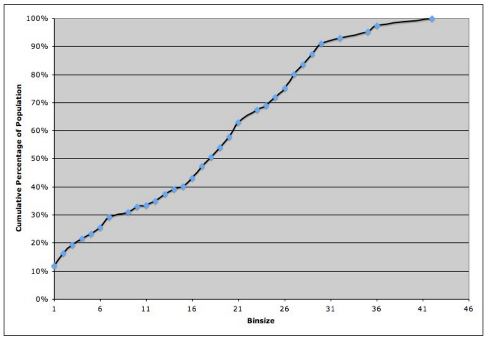
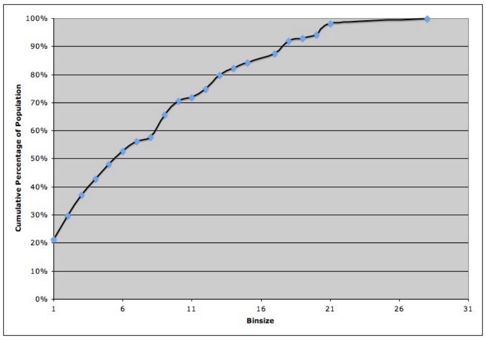
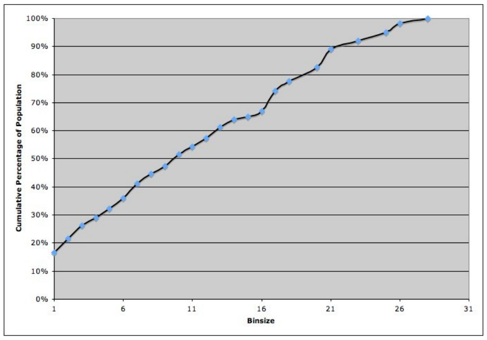
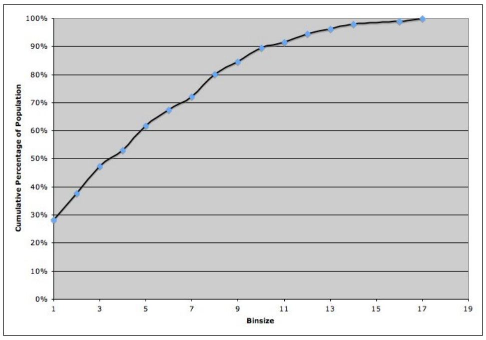
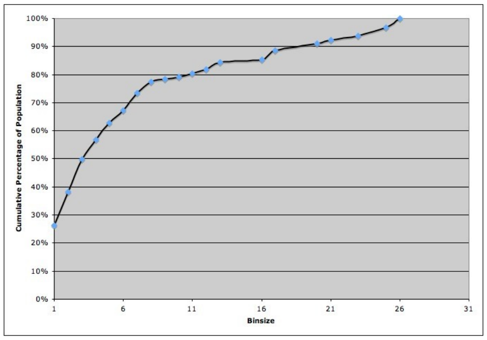
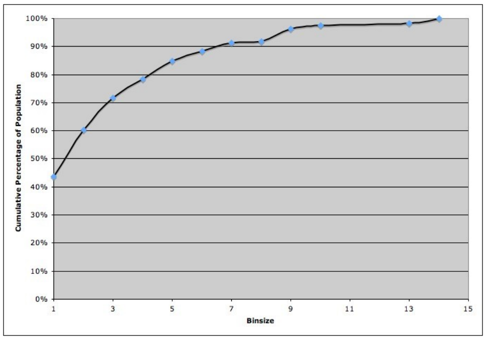
ZIP3	 12% unique (193 Clients)	 21% (346 Clients)
ZIP4	 17% unique (267 Clients)	 28% (455 Clients)
ZIP5	 26% unique (423 Clients)	 44% unique (704 Clients)
	Gender	**Gender, Race, Ethnicity**
	Age	

Figure 96. Comparison of cumulative binsize distributions for combined demographics, age and ZIP aggregations in Iowa de-duplicated results. Counts based on 1614 clients in the De-duplicated Demographics Database. Categories of ZIP are 5-digits (ZIP5), the first 4 digits (ZIP4) and the first 3 digits (ZIP3). Age computed as a 2-year range using year of birth.

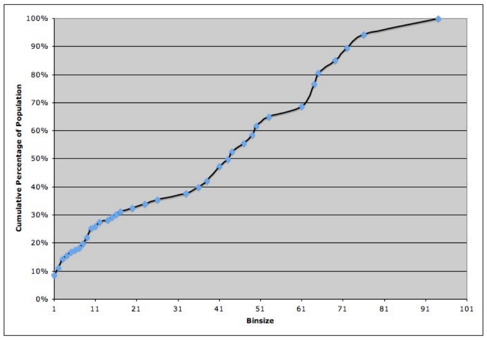
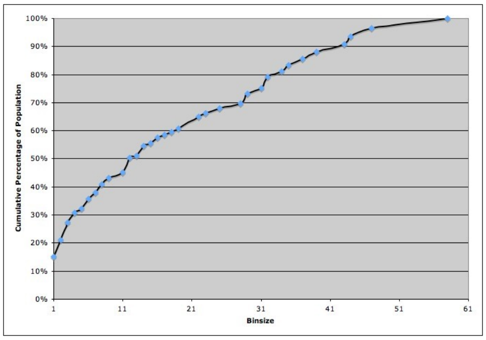
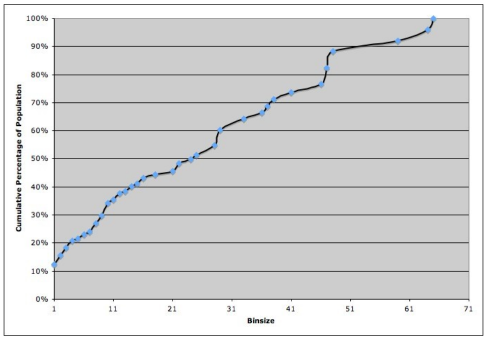
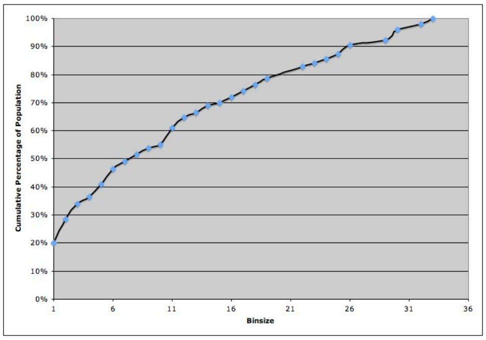
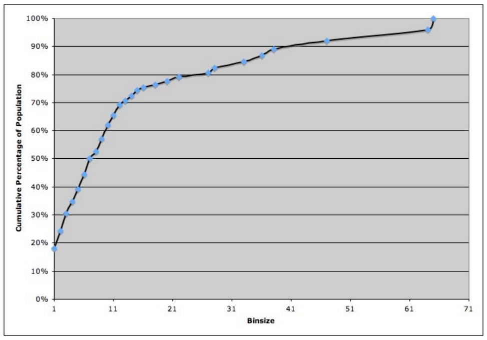
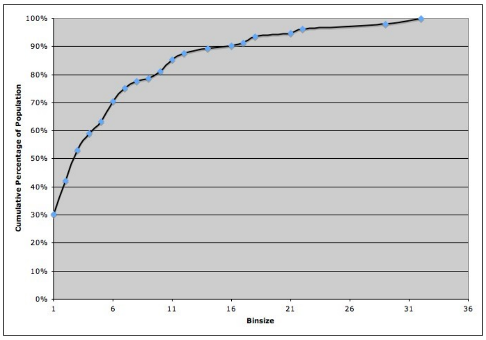
ZIP3	 9% unique (139 Clients)	 15% (245 Clients)
ZIP4	 12% unique (201 Clients)	 20% (326 Clients)
ZIP5	 18% unique (294 Clients)	 30% unique (488 Clients)
	Gender	**Gender, Race, Ethnicity**
	5-year Age Ranges	

Figure 97. Comparison of cumulative binsize distributions for combined demographics, 5-year age ranges and ZIP aggregations in Iowa de-duplicated results. Counts based on 1614 clients in the De-duplicated Demographics Database. Categories of ZIP are 5-digits (ZIP5), the first 4 digits (ZIP4) and the first 3 digits (ZIP3).

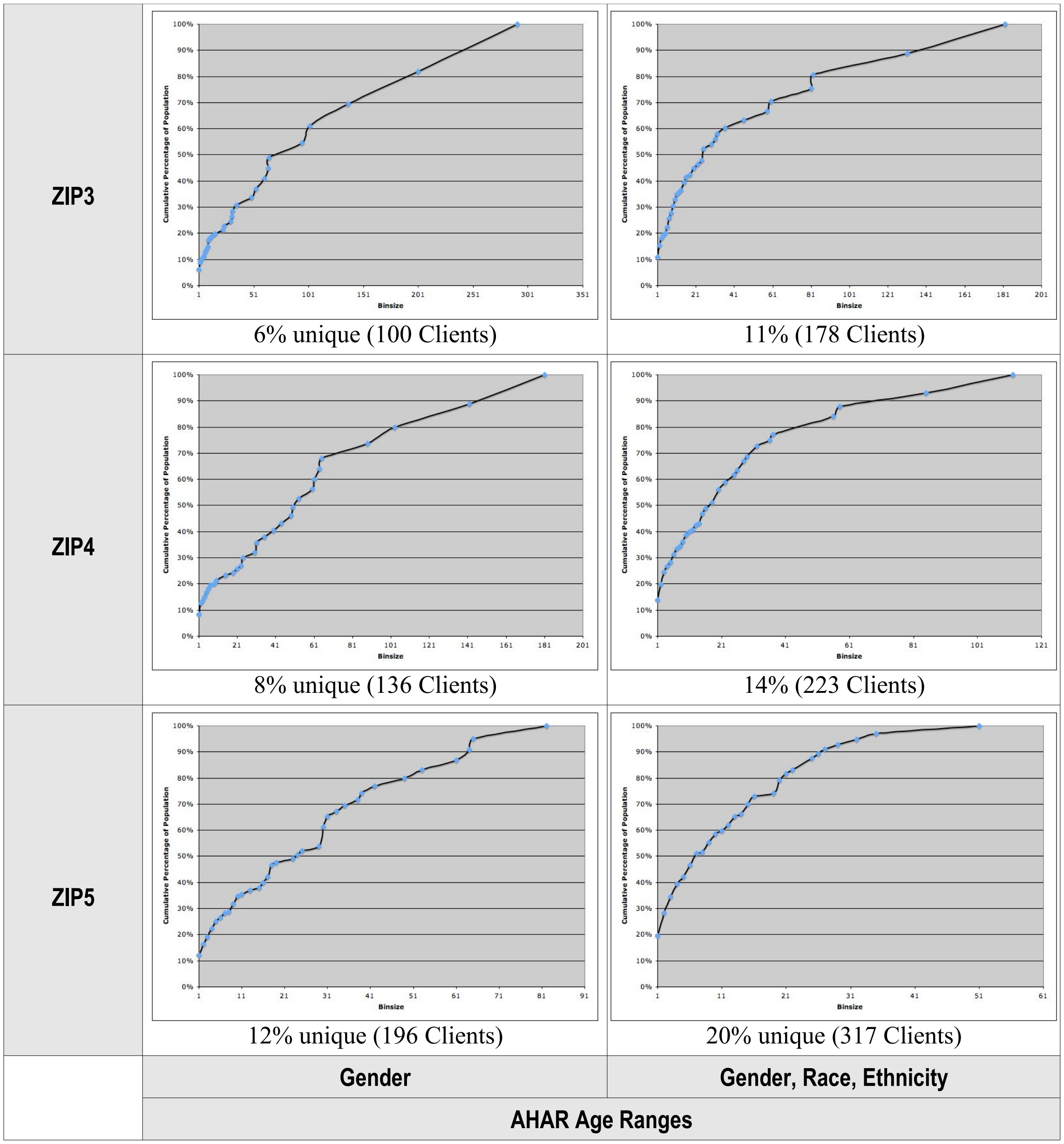
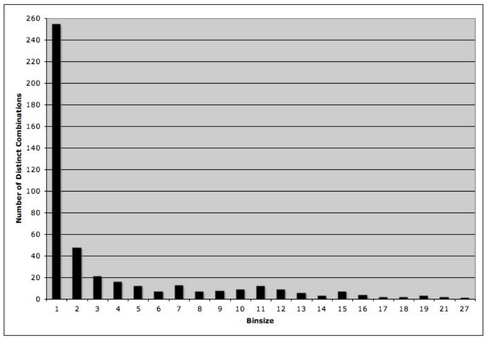
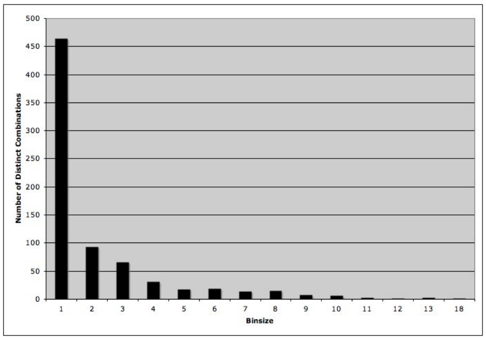
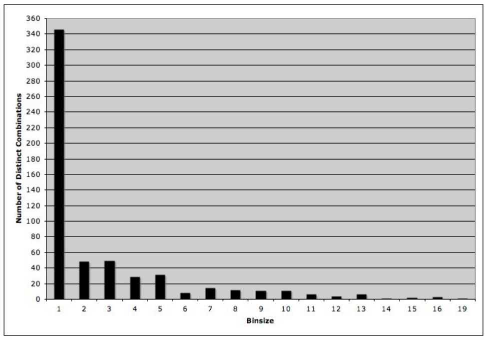
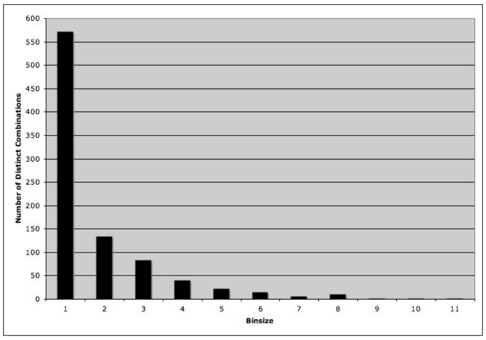


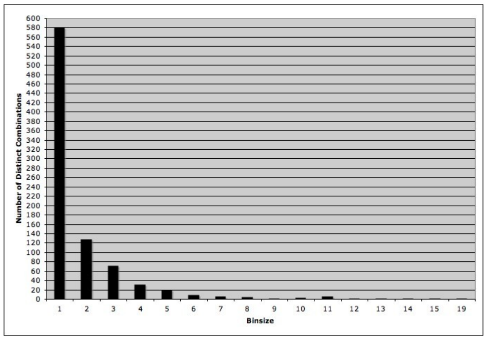
Figure 98. Comparison of cumulative binsize distributions for combined demographics, AHAR age ranges and ZIP aggregations in Iowa de-duplicated results. Counts based on 1614 clients in the De-duplicated Demographics Database. Categories of ZIP are 5-digits (ZIP5), the first 4 digits (ZIP4) and the first 3 digits (ZIP3). AHAR age ranges: Under 1, 1 to 5, 6 to 12, 13 to 17, 18 to 30, 31 to 50, 51 to 61, and 62 and above.

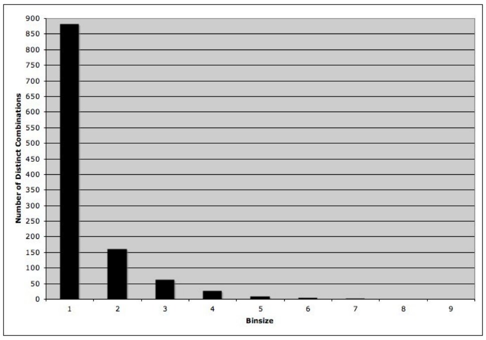
<table>
<tr>
<td><b>ZIP3</b></td>
<td>


16% unique (255 Clients)
</td>
<td>


29% (464 Clients)
</td>
</tr>
<tr>
<td><b>ZIP4</b></td>
<td>


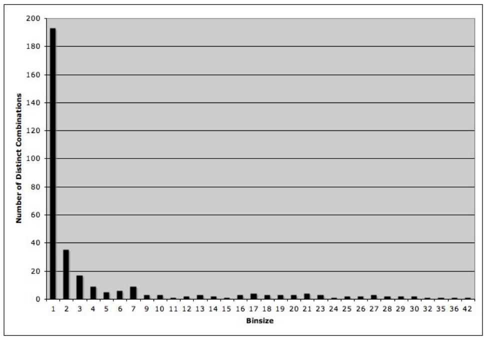
21% unique (346 Clients)
</td>
<td>


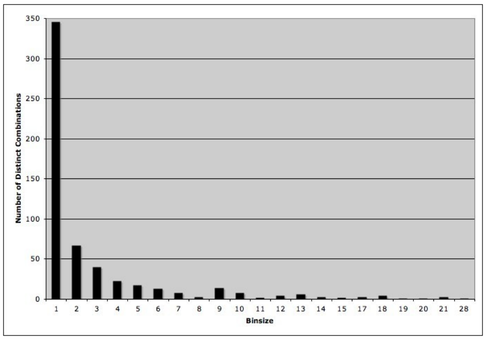
35% (571 Clients)
</td>
</tr>
<tr>
<td><b>ZIP5</b></td>
<td>


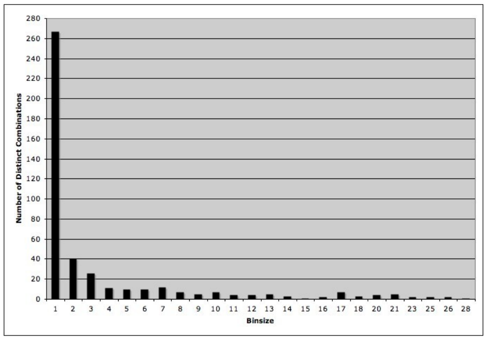
36% unique (580 Clients)
</td>
<td>


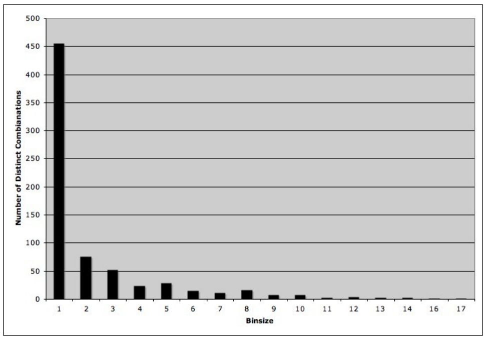
55% unique (882 Clients)
</td>
</tr>
<tr>
<td rowspan="2"></td>
<td><b>Gender</b></td>
<td><b>Gender, Race, Ethnicity</b></td>
</tr>
<tr>
<td colspan="2"><b>Year of Birth</b></td>
</tr>
</table>

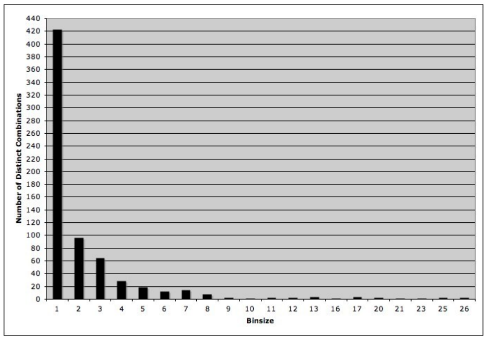
Figure 99. Comparison of binsize distributions for combined demographics, year of birth and ZIP aggregations in Iowa de-duplicated results. Counts based on 1614 clients in the De-duplicated Demographics Database. Categories of ZIP are 5-digits (ZIP5), the first 4 digits (ZIP4) and the first 3 digits (ZIP3).

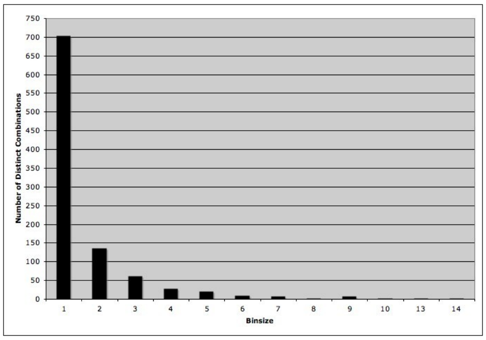
<table>
<tr>
<td><b>ZIP3</b></td>
<td>

12% unique (193 Clients)</td>
<td>

21% (346 Clients)</td>
</tr>
<tr>
<td><b>ZIP4</b></td>
<td>

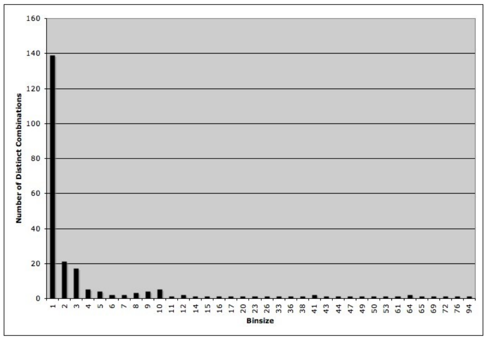
17% unique (267 Clients)</td>
<td>

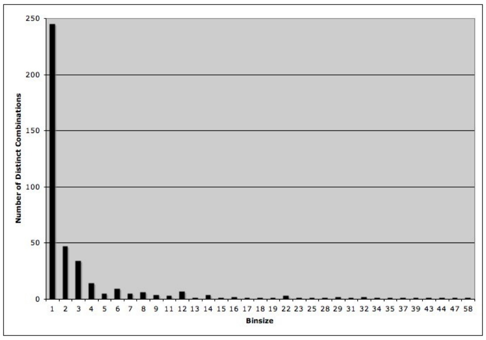
28% (455 Clients)</td>
</tr>
<tr>
<td><b>ZIP5</b></td>
<td>

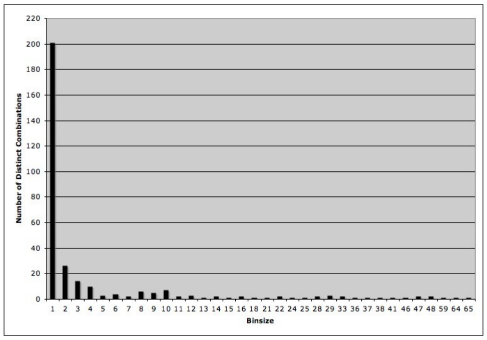
26% unique (423 Clients)</td>
<td>

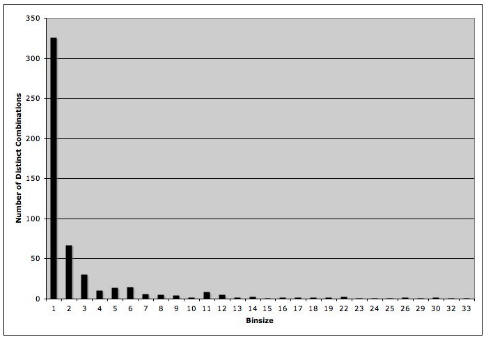
44% unique (704 Clients)</td>
</tr>
<tr>
<td></td>
<td><b>Gender</b></td>
<td><b>Gender, Race, Ethnicity</b></td>
</tr>
<tr>
<td></td>
<td colspan="2"><b>Age</b></td>
</tr>
</table>

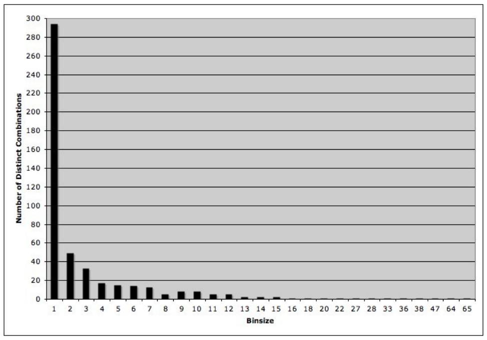
Figure 100. Comparison of binsize distributions for combined demographics, age and ZIP aggregations in Iowa de-duplicated results. Counts based on 1614 clients in the De-duplicated Demographics Database. Categories of ZIP are 5-digits (ZIP5), the first 4 digits (ZIP4) and the first 3 digits (ZIP3). Age computed as a 2-year range using year of birth.

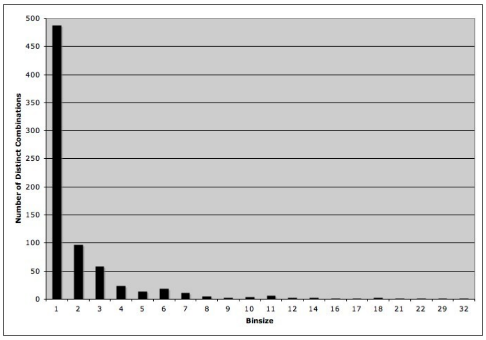
<table>
<tr>
<td><b>ZIP3</b></td>
<td>

9% unique (139 Clients)</td>
<td>

15% (245 Clients)</td>
</tr>
<tr>
<td><b>ZIP4</b></td>
<td>

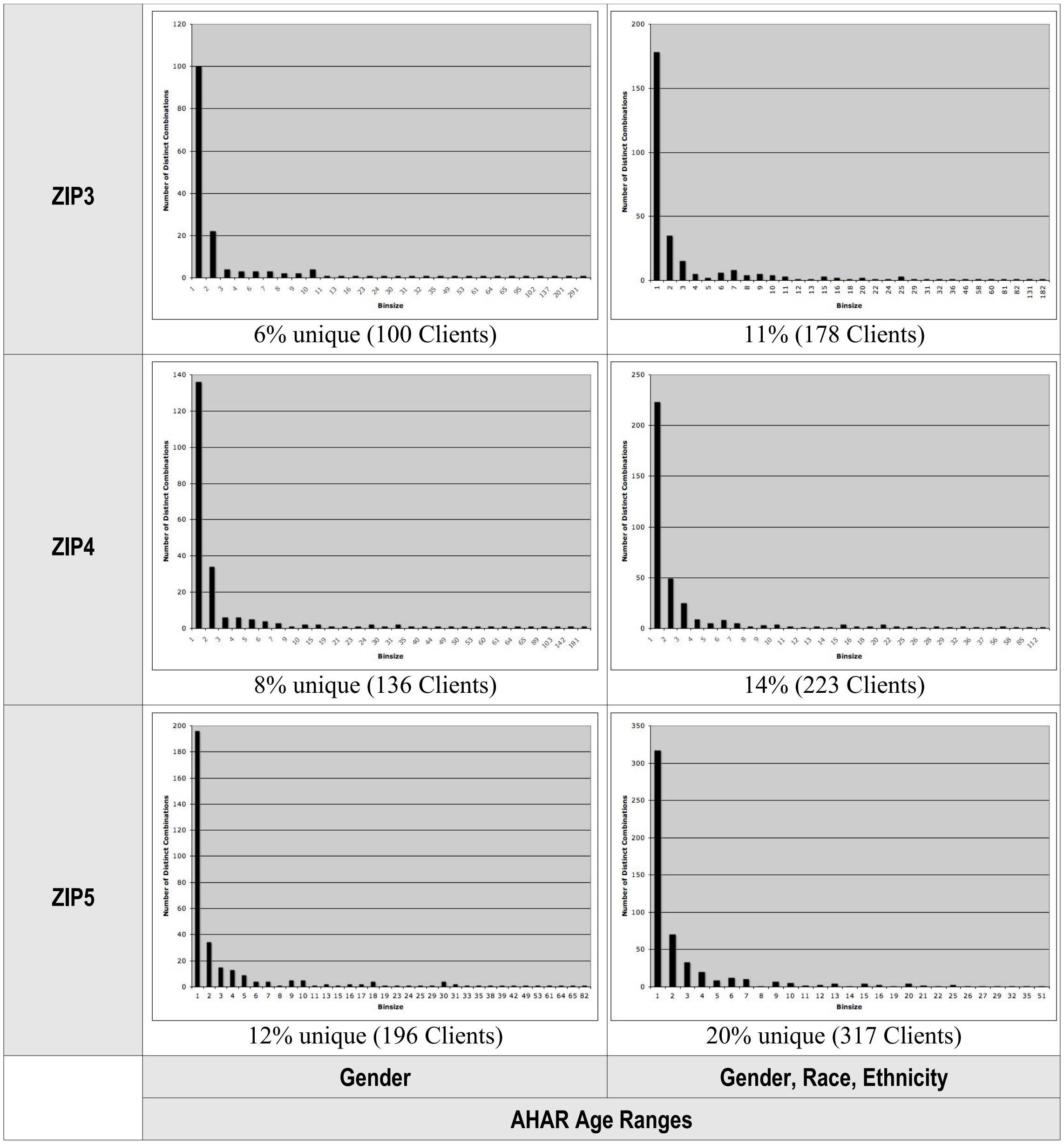
12% unique (201 Clients)</td>
<td>

20% (326 Clients)</td>
</tr>
<tr>
<td><b>ZIP5</b></td>
<td>

18% unique (294 Clients)</td>
<td>

30% unique (488 Clients)</td>
</tr>
<tr>
<td></td>
<td><b>Gender</b></td>
<td><b>Gender, Race, Ethnicity</b></td>
</tr>
<tr>
<td></td>
<td colspan="2"><b>5-year Age Ranges</b></td>
</tr>
</table>

Figure 101. Comparison of binsize distributions for combined demographics, 5-year age ranges and ZIP aggregations in Iowa de-duplicated results. Counts based on 1614 clients in the De-duplicated Demographics Database. Categories of ZIP are 5-digits (ZIP5), the first 4 digits (ZIP4) and the first 3 digits (ZIP3).



ZIP3: 6% unique (100 Clients) | 11% (178 Clients)

ZIP4: 8% unique (136 Clients) | 14% (223 Clients)

ZIP5: 12% unique (196 Clients) | 20% unique (317 Clients)

Figure 102. Comparison of binsize distributions for combined demographics, AHAR age ranges and ZIP aggregations in Iowa de-duplicated results. Counts based on 1614 clients in the De-duplicated Demographics Database. Categories of ZIP are 5-digits (ZIP5), the first 4 digits (ZIP4) and the first 3 digits (ZIP3). AHAR age ranges: Under 1, 1 to 5, 6 to 12, 13 to 17, 18 to 30, 31 to 50, 51 to 61, and 62 and above.

11.3 Re-identification of Universal Data Elements

As stated earlier, PrivaMix only provides guaranteed privacy protection for UID creation and use in de-duplicating. The privacy of the data elements associated with the UIDs are beyond the scope of the PrivaMix Demonstration System. However, the next section (Section 12) examines some possible ways for PrivaMix to provide privacy protection to Universal Data Elements by expanding its post processing. In the absence of such a remedy, the Universal Data Elements themselves must be altered to thwart data linking (Section 4.2).

This section briefly discusses three re-identification strategies: (1) linking on demographic data; and (2) trail re-identification by HMIS.

11.3.1. Data linkage using demographic fields

Section 4.5 examined the identifiability of the Universal Data Elements. These fields currently include the full month, day and year of birth and the 5-digit ZIP. Figure 13 shows the identifiability of combinations of aggregations of these values. It reports that {date of birth, gender, 5-digit ZIP} uniquely identifies 97% of the U.S. Population. Even changing date of birth to year of birth drastically reduces the identifiability. Figure 13 shows that {year of birth, gender, 5-digit ZIP} uniquely identifies 0.04% of the U.S. Population.

Section 11.3 revealed a somewhat substantial number of unique combinations of values appearing in demographic fields. Figure 89 shows that 36% of Clients in the De-duplicated Demographics Database had unique combinations of {year of birth, gender, 5-digit ZIP}. At first glance, this seems to contradict the identifiability rate mentioned above of 0.04%. That's because having a unique combination of demographic values does not necessarily make the Client identifiable. Successful re-identification requires another dataset on which to link to actually re-identify the Client. If a re-identification attempt only has access to data on the general population, such as a voter list, then the likelihood of re-identification using {year of birth, gender, ZIP} is 0.04%. However, if a re-identification attempt has access to the HMIS, the likelihood of a re-identification using {year of birth, gender, ZIP} is 36%. The HMIS seems to hold sufficient data that the percentage of uniquely occurring combinations of demographic values approximates their likelihood of unique re-identification of Clients.

11.3.2. Data linkage using exact service dates

Another strategy for an HMIS to re-identify Clients in de-duplicated data containing the Universal Data Elements is to exploit the service dates, which can uniquely combine with even the most general demographics, to re-identify Clients.

When a Client receives a service not limited to domestic violence homeless shelters, the Client's explicitly identifying information appears alongside that record in the HMIS. When the HMIS de-duplicates with Shelters, the record of a Client at a Shelter and a record of that same client receiving a non-DV service are related.

If the HMIS has access to the de-duplicated results, the demographic, entry date, and exit date fields may combine sufficient to reliably re-identify Clients by matching service dates. Because the non-DV record has the Client's name, the HMIS learns the client's DV information.

A possible remedy is to provide number of days of service or time ranges (e.g., overnight, a week or less, a month or less, more than a month) and not the actual dates of service.

11.3.3. Trail re-identification using exact service sequence

Another strategy for an HMIS to re-identify Clients in de-duplicated data containing the Universal Data Elements is to exploit the sequence of services received. The service dates provide a longitudinal record of services received by a Client. This longitudinal record poses a linkage threat.

Because the exact entry and exit dates appear in the Universal Data Elements, the Client has a longitudinal record of services over time. The sequence of provided services is likely unique to each Client.

A possible remedy is to provide number of days of service or time ranges (e.g., overnight, a week or less, a month or less, more than a month) and not the actual dates of service.

11.4 Changes to Universal Data Elements

Below are recommendations related to demographics appearing in the Universal Data Elements.

Recommendation #34: The AHAR does not require the demographic specificity currently found in the Universal Data Elements. More general values can be shared without any loss to reporting ability. Therefore, the Universal Data Elements should be revised to reduce the likelihood of recognition by the intimate stalker and/or data linkage threats by using the most general values possible.

Recommendation #35: The date of birth field should minimally be an age range. In fact, a Client may have more than one kind of age range specification. For example, there may be a data element related to 5-year age ranges, and another related to AHAR ranges (under 1, 1 through 5, 6 through 12, 13 through 17, 18 through 30, 31 through 50, 51 through 61, and 62 and over), enabling more reporting uses of the resulting data.

Recommendation #36: The ZIP of last residence field should be changed to either report the first 3 digits of ZIP, or even better, be changed to be a boolean flag denoting whether the Client's last residence was within the geography covered by the Planning Office or not. If the first 3 digits of ZIP are used, then only those values local to the Planning Office need be recorded. Clients from outside the local area would just have a special value, like 999, in order to prevent them appearing as unique outliers.

Recommendation #37: PIN should be removed. The Shelter should not provide its internal unique number. Instead, the Shelter should maintain an exact copy of the data provided so that records can be referred to in discussion with the Planning Office by the place (or row) in which the record appears.

Recommendation #38: Consider removing Race and Ethnicity. Experimental results showed that the addition of these fields increase risks to re-identification.

Recommendation #39: Shelters should consider renumbering Household identification numbers from 1 to the last household, prior to forwarding the information to the Planning Office. This makes sure the household identification number itself cannot be the basis for linking.

Recommendation #40: Replace the exact service dates (Program Entry Date and Program Exit Date) with number of days of service or with time periods (e.g., overnight, 2-14 days, 15-30 days, 30 plus days).

Recommendation #41: More sensitive data elements (such as first name, Social Security number, or full date of birth) may still be collected by Shelters in order to produce a useful UID. However, those values should continue to not be forwarded to the Planning Office as part of the Universal Data Elements.

12. Privacy Assurance Using PrivaMix

In order to prevent collusion when the Planning Office and the HMIS consist of the same personnel, it is necessary to use additional privacy safeguards, beyond those for protecting UIDs (e.g. PrivaMix) and beyond merely changing the Universal Data Elements. Remedies involve expanding the post-processing done by PrivaMix so that the final dataset made available to the Planning Office contains either aggregate (not Client-level data) or provably anonymized Client-level data.

While PrivaMix guarantees privacy protection for UID creation and use in de-duplicating, linking vulnerabilities currently remain in the de-duplicated Universal Data Elements (Section 11). Problems stem from the selection of which data elements to associate with UIDs, and not from the UIDs themselves. Changes to the Universal Data Elements can help (Section 11), but such changes seem unable to be wholly satisfactory without effecting the usefulness of the de-duplicated data to the AHAR.

A PrivaMix System can anonymize de-duplicated results prior to forwarding data to the Planning Office. The anonymizaed data will not be vulnerable to linking, even if the Planning Office and HMIS collude.

At present, the PrivaMix Demonstration System, as used in the Iowa Experiment, de-duplicates Client information and then passes values associated with each UID to the Planning Office "as is." Instead of merely forwarding those values, a PrivaMix System could anonymize those data elements and then forward the anonymized results to the Planning Office.

There are numerous way for a PrivaMix System to perform anonymization. These include: replacing client-level results with pivot tables that show aggregate count information for combinations of data elements; replacing client-level data with an overall final report (e.g., the AHAR itself); or, provably anonymizing client-level data by automatically suppressing and generalizing values as needed. Each of these approaches can provide sufficient privacy protection, by replacing client-specific results with appropriately generalized ones. The result is privacy protection, even against data linking, and accurate de-duplicated results for the AHAR.

A way to thwart HMIS linking of Universal Data Elements without expanding PrivaMix is to have all clients, whether they be domestic violence clients or not, use the same privacy protections of the domestic violence clients. Then, the HMIS itself would lack explicit identifiers of clients, making linking less useful. The viability of this option in terms of the overall utility of the HMIS is beyond the scope of this writing.

Below are recommendations based on the discussion above.

Recommendation #43: *In order to prevent collusion when the Planning Office and the HMIS consist of the same personnel, it is necessary to use additional privacy safeguards, beyond those for protecting UIDs (e.g. PrivaMix) and beyond merely changing the Universal Data Elements. It is necessary to make sure the HMIS cannot link the Universal Data Elements to other service information contained in the HMIS.*

Recommendation #44: Add post de-duplication anonymization to a PrivaMix System to make sure data provided to the Planning Office is not vulnerable to linking, even if the Planning Office and HMIS collude. The Planning Office receives provably anonymized de-duplicated results.

Recommendation #45: Consider having the final results be aggregate data only. Instead of Client-level data, a PrivaMix System can alternatively provide aggregate de-duplicated count distributions denoting how many Clients matched particular characteristics. An example of a count distribution are counts by age ranges. Distributions can involve more than one field to get more specific data.

Recommendation #46: Consider having the final results be the AHAR report itself. Instead of Client-level data, a PrivaMix System can alternatively provide the AHAR to the Planning Office.

Recommendation #47: Consider having the final results be anonymized Client-level data. Anonymized Client-level data generalizes or suppresses values, as needed, to protect privacy. Formal protection models identify which values to generalize or suppress from the resulting dataset so that each record ambiguously relates to a minimum number of people [30][31]. For example, if a 80 year old woman is an outlier in the data because of her age, either her age would be removed from the data or generalized to a category having more people, such as "50 plus" as appropriate value given the other ages appearing in the data.

In conclusion, PrivaMix provides an effective and accurate privacy-preserving means for constructing and de-duplicating UIDs. However, additional care with the Universal Data Elements must be taken to properly protect against unwanted data linkage with the HMIS. The problem is not with the UIDs but with the selection of data elements associated with the UIDs. A solution is to enhance a PrivaMix System to anonymize de-duplicated Client-level data and then forward the anonymized results to the Planning Office.

Acknowledgements

The author thanks Michael Roanhouse, Mary Joel Holin, Julie Hovden, Brian Sokol, Michelle Hayes, Julie Eberbach, David Eberbach, Eileen Mitchell, Jody Miller, Cindy Southworth, and participating stakeholders for the opportunity to do this work. This research was conducted on an independent, scientific basis. The work was funded in part by Abt Associates as part of a grant from the United States Department of Housing and Urban Development (HUD). The views and results expressed herein are solely those of Dr. Sweeney and do not necessarily represent those of HUD or Abt Associates.

References

1 U.S. Department of Housing and Urban Development. Homeless Management Information Systems (HMIS); Data and Technical Standards Final Notice. *Federal Register*, Vol. 69, No. 146, July 30, 2004, p. 45888-45934.

2 U.S. Department of Housing and Urban Development. *Homeless Management Information Systems (HMIS) Data and Technical Standards Final Notice; Clarification and Additional Guidance on Special Provisions for Domestic Violence Provider Shelters.* Docket No. FR 4848-N-O3. August 30, 2004.

3 U.S. Department of Housing and Urban Development. Emergency Shelter Grants Allocation History. www.hud.gov/utilities/intercept.cfm?/offices/cpd/homeless/budget/esghistory.pdf as of September 2005.

4 Northeast Ohio Coalition for the Homeless. *Overflowing Shelters: a history and recommended solutions*. April 9, 2005. www.neoch.org/what_to_do_overflowing.htm as of September 2005.

5 U.S. Conference of Mayors, A Status Report on Hunger and Homelessness in America's Cities 2001. www.usmayors.org/uscm/hungersurvey/2001/hungersurvey2001.pdf as of Sept 2005.

6 Markee, P. *Average Daily Census of Homeless Children and Adults Residing in the New York City Municipal Shelter System*. Coalition for the Homeless on behalf of New York City Department of Homeless Services and Human Resources Administration, May, 2002.

7 New York City Independent Budget Office. *Give 'Em Shelter: Various City Agencies Spend Over $900 Million on Homeless Services*. Fiscal Brief, March 2002

8 Conference Report (H.R. Report 107-272) for the Fiscal Year 2002 HUD Appropriations Act (Public Law 107-73).

9 Senate Committee Report 107-43 for the Fiscal Year 2002 HUD Appropriations Act (Public Law 107-43).

10 Conference Report (H.R. Report 106-988) for the Fiscal Year 2001 HUD Appropriations Act (Public Law 106-377).

11 U.S. Bureau of the Census. *1990 Collection and Processing Procedures (Appendix D)* . CD-ROM Technical Documentation Project. University of Michigan. February 1998. www.lib.umich.edu/govdocs/cicdoc/cen90app/append_d.htm as of September 2005.

12 U.S. Bureau of the Census. *1996 National Survey of Homeless Assistance Providers and Clients*. Washington: 1996. www.census.gov/prod/www/nshapc/NSHAPC4.html as of September 2005.

13 M. Burt and L. Aron. *America's Homeless II: Population and Services*. Urban Institute. Washington: 2000. www.urban.org/UploadedPDF/900344_AmericasHomelessII.pdf as of Sept 2005

14 Electronic Privacy Information Center. Comments to HUD on the Matter of HMIS. Sept 2003.

15 The National Network to End Domestic Violence. Comments to HUD on the Matter of HMIS. Sept. 2003

16 National Center for Victims of Crime. Domestic Violence. As of Sept 2005, www.ncvc.org/ncvc/main.aspx?dbName=DocumentViewer&DocumentID=32347

17 U.S. Department of Justice. *Violence by Intimates: analysis of data on crimes by current or former spouses, boyfriends, and girlfriends.* NCJ-167237. March 1998.

18 S. Catania. No safe haven. *Mother Jones*, July/August 2005.

19 National HMIS TA Initiative Documents: AHAR Super Table Shells. As of Sept 2005, www.hmis.info/ta_resources_data.asp?topic_id=11

20 Privacert, Inc. *The Privacert Risk Assessment Server*. Available at www.privacert.com as of Sept 2005. Originally designed and developed by L. Sweeney.

21 Pfleeger, C. *Security in Computing*. Prentice-Hall. Upper Saddle River: 1997

22 Stinson, D. *Cryptography: Theory and Practice*. CRC Press. New York: 1995

23 Ratha, N. and Bolle, R. Automatic Fingerprint Recognition Systems. Springer-Verlag. New York: 2004

24 Russell, R. Soundex. U.S. Patent 1,261,167 April 2, 1918.

25 Record Linkage Techniques -- 1997: Proceedings of an International Workshop and Exposition. National Research Council, 1999.

26 Sweeney, L. *Inconsistent Hashing and the Notion of Single-Use Identifying Numbers*. Carnegie Mellon University, School of Computer Science, Data Privacy Lab White Paper Series LIDAP-WP13. Pittsburgh, PA: 2005.

27 Edo-Eket, S. and Sweeney, L. *Detecting Bio-Terrorist Attacks and Naturally Occurring Outbreaks Over a Distributed Network While Protecting Privacy and Confidentiality: the PrivaSum Protocol*. Carnegie Mellon University, School of Computer Science, Technical Report CMU-ISRI-04-111.

28 J. Benaloh and M. de Mare. One-way accumulators: a decentralized alternative to digital signatures. In *Proceedings of Advances in Cryptology - EUROCRYPT '93, Lecture Notes in Computer Science*, v 765, pages 274-285, Lofthus, Norway, 1994.

29 Sweeney, L. and Shamos, M. *A Multiparty Computation for Randomly Ordering Players and Making Random Selections*. Carnegie Mellon University, School of Computer Science, Technical Report, CMU-ISRI-04-126. Pittsburgh: July 2004. privacy.cs.cmu.edu/dataprivacy/projects/randomorder/index.html

30 Sweeney, L. k-anonymity: a model for protecting privacy. I*nternational Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10 (5), 2002; 557-570. privacy.cs.cmu.edu/people/sweeney/kanonymity.html.

31 Sweeney, L. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10 (5), 2002; 571-588. privacy.cs.cmu.edu/people/sweeney/kanonymity2.html.

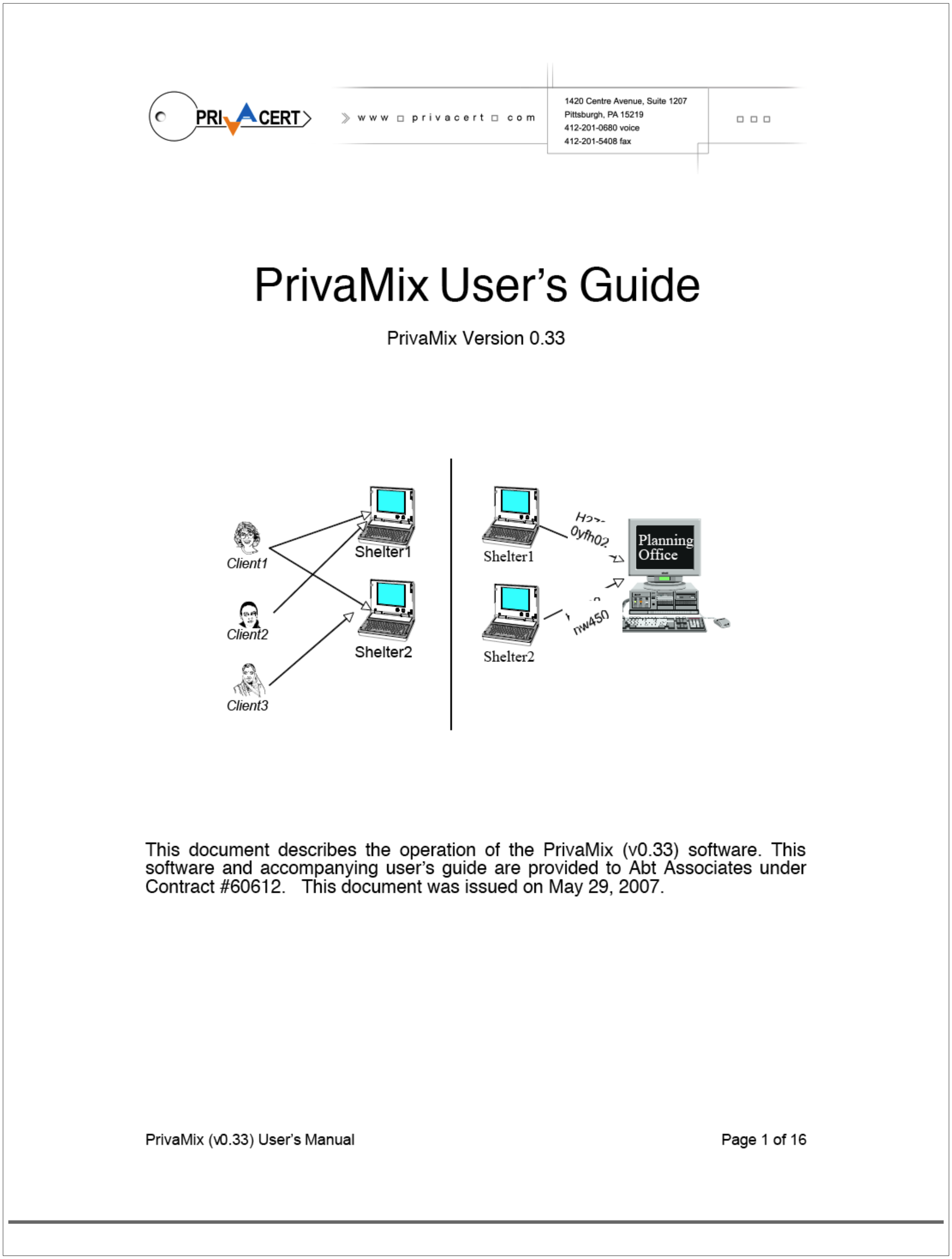
32 Sweeney, L. The Search for a P3Tracker Hash Function: a research notebook. Carnegie Mellon University. LIDAP Working Paper 31. Pittsburgh: April 2006.

33 Russell, R. *Soundex*. U.S. Patent 1261167. April 2, 1918.

34 Sokol, B. *PrivaMix Proof-of-Concept Report*. Abt Associates. July 2007.

Appendix A

PrivaMix User's Guide

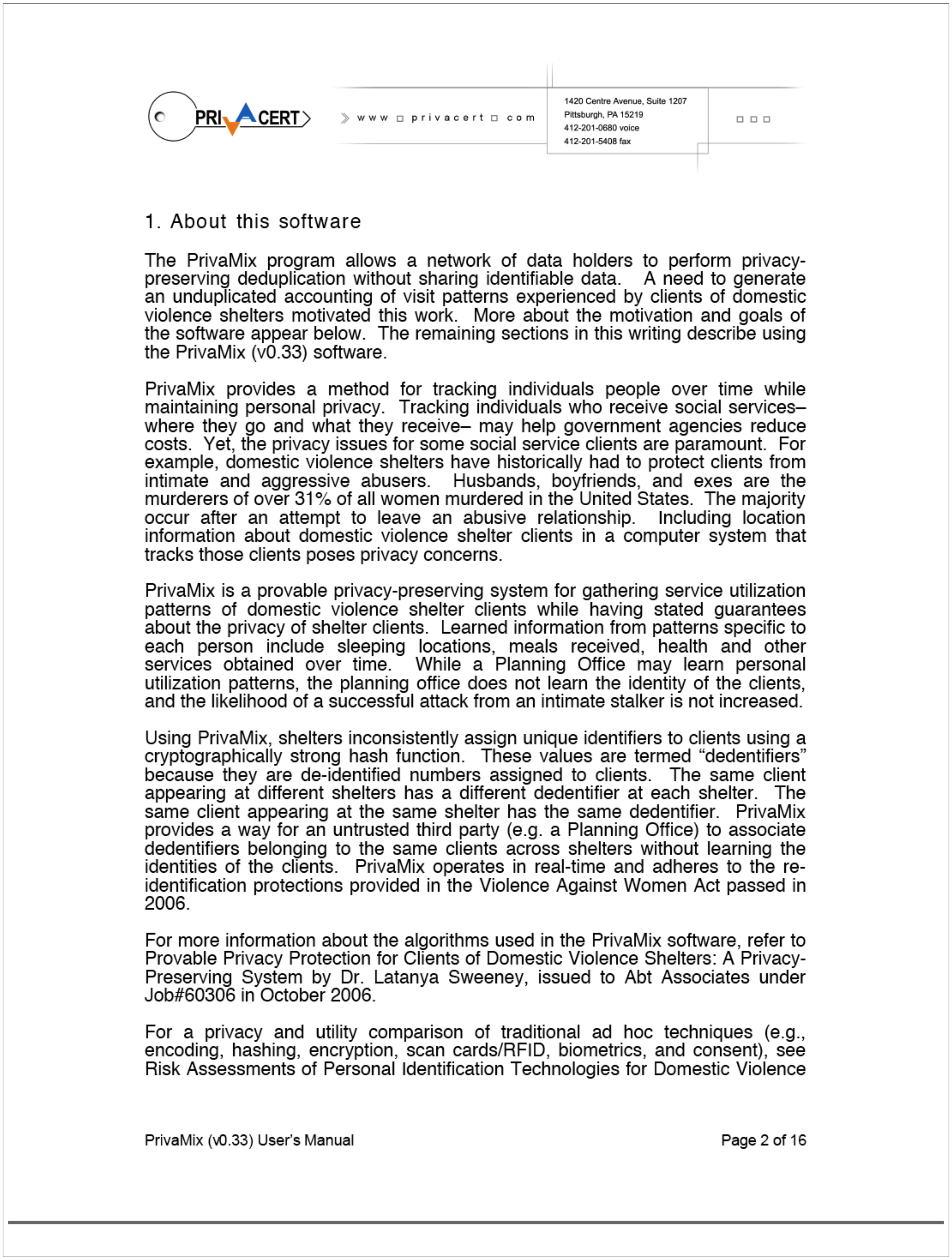


PRIVACERT
www.privacert.com
1420 Centre Avenue, Suite 1207
Pittsburgh, PA 15219
412-201-0680 voice
412-201-5408 fax

PrivaMix User's Guide

PrivaMix Version 0.33

This document describes the operation of the PrivaMix (v0.33) software. This software and accompanying user's guide are provided to Abt Associates under Contract #60612. This document was issued on May 29, 2007.



1. About this software

The PrivaMix program allows a network of data holders to perform privacy-preserving deduplication without sharing identifiable data. A need to generate an unduplicated accounting of visit patterns experienced by clients of domestic violence shelters motivated this work. More about the motivation and goals of the software appear below. The remaining sections in this writing describe using the PrivaMix (v0.33) software.

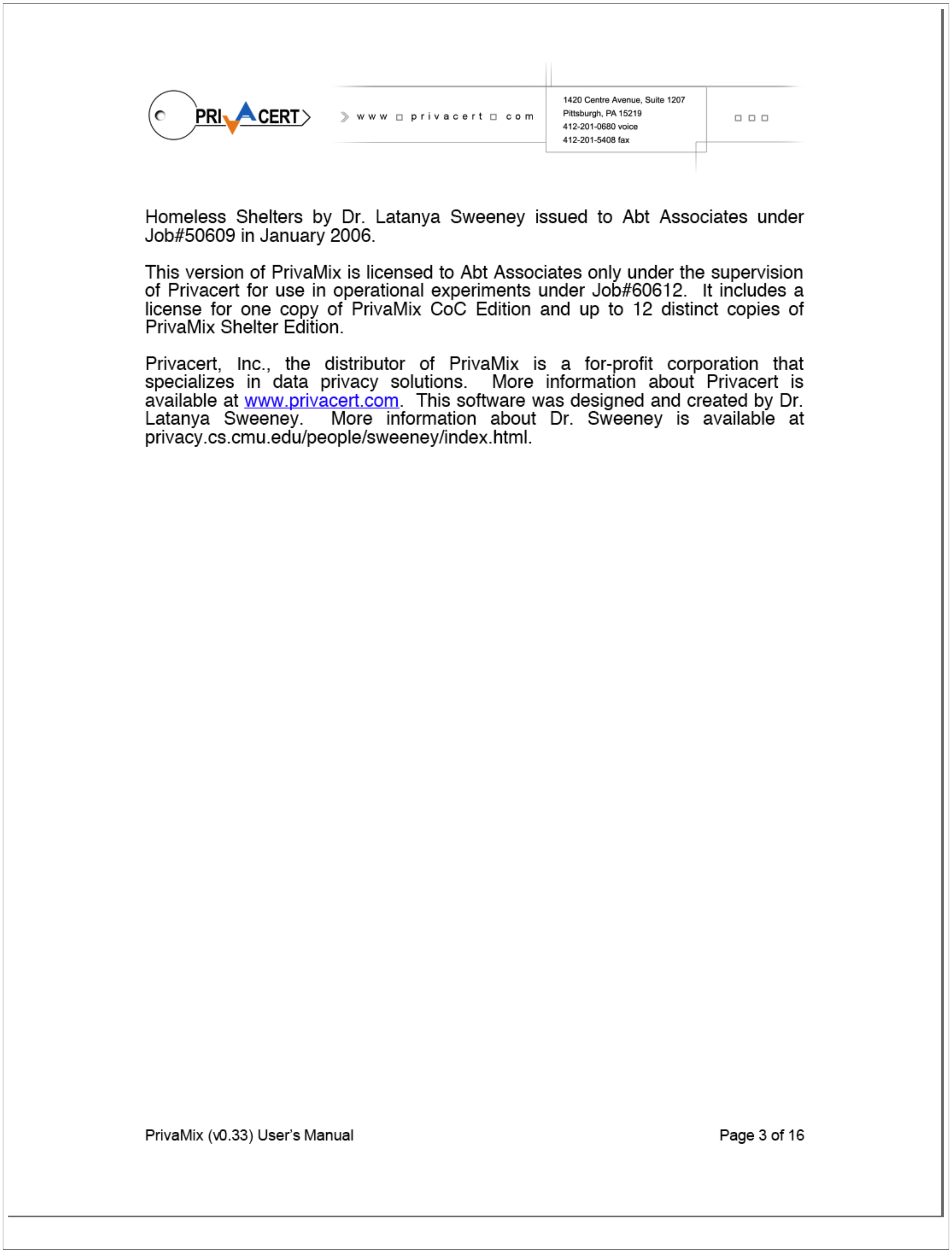
PrivaMix provides a method for tracking individuals people over time while maintaining personal privacy. Tracking individuals who receive social services– where they go and what they receive– may help government agencies reduce costs. Yet, the privacy issues for some social service clients are paramount. For example, domestic violence shelters have historically had to protect clients from intimate and aggressive abusers. Husbands, boyfriends, and exes are the murderers of over 31% of all women murdered in the United States. The majority occur after an attempt to leave an abusive relationship. Including location information about domestic violence shelter clients in a computer system that tracks those clients poses privacy concerns.

PrivaMix is a provable privacy-preserving system for gathering service utilization patterns of domestic violence shelter clients while having stated guarantees about the privacy of shelter clients. Learned information from patterns specific to each person include sleeping locations, meals received, health and other services obtained over time. While a Planning Office may learn personal utilization patterns, the planning office does not learn the identity of the clients, and the likelihood of a successful attack from an intimate stalker is not increased.

Using PrivaMix, shelters inconsistently assign unique identifiers to clients using a cryptographically strong hash function. These values are termed “dedentifiers” because they are de-identified numbers assigned to clients. The same client appearing at different shelters has a different dedentifier at each shelter. The same client appearing at the same shelter has the same dedentifier. PrivaMix provides a way for an untrusted third party (e.g. a Planning Office) to associate dedentifiers belonging to the same clients across shelters without learning the identities of the clients. PrivaMix operates in real-time and adheres to the re-identification protections provided in the Violence Against Women Act passed in 2006.

For more information about the algorithms used in the PrivaMix software, refer to Provable Privacy Protection for Clients of Domestic Violence Shelters: A Privacy-Preserving System by Dr. Latanya Sweeney, issued to Abt Associates under Job#60306 in October 2006.

For a privacy and utility comparison of traditional ad hoc techniques (e.g., encoding, hashing, encryption, scan cards/RFID, biometrics, and consent), see Risk Assessments of Personal Identification Technologies for Domestic Violence



www □ privacert □ com

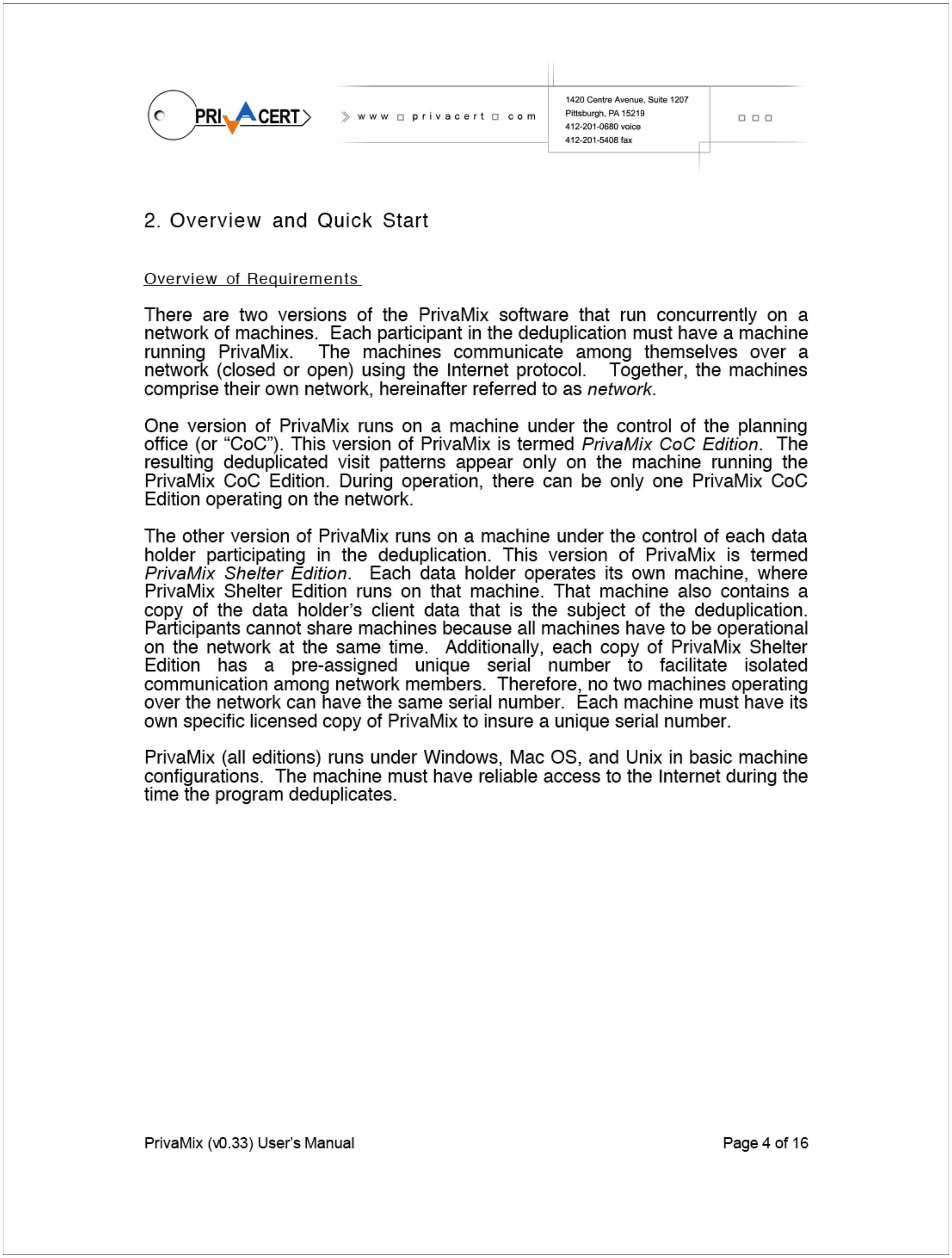
1420 Centre Avenue, Suite 1207
Pittsburgh, PA 15219
412-201-0680 voice
412-201-5408 fax

Homeless Shelters by Dr. Latanya Sweeney issued to Abt Associates under Job#50609 in January 2006.

This version of PrivaMix is licensed to Abt Associates only under the supervision of Privacert for use in operational experiments under Job#60612. It includes a license for one copy of PrivaMix CoC Edition and up to 12 distinct copies of PrivaMix Shelter Edition.

Privacert, Inc., the distributor of PrivaMix is a for-profit corporation that specializes in data privacy solutions. More information about Privacert is available at www.privacert.com. This software was designed and created by Dr. Latanya Sweeney. More information about Dr. Sweeney is available at privacy.cs.cmu.edu/people/sweeney/index.html.

PrivaMix (v0.33) User's Manual Page 3 of 16



2. Overview and Quick Start

Overview of Requirements

There are two versions of the PrivaMix software that run concurrently on a network of machines. Each participant in the deduplication must have a machine running PrivaMix. The machines communicate among themselves over a network (closed or open) using the Internet protocol. Together, the machines comprise their own network, hereinafter referred to as *network*.

One version of PrivaMix runs on a machine under the control of the planning office (or "CoC"). This version of PrivaMix is termed *PrivaMix CoC Edition*. The resulting deduplicated visit patterns appear only on the machine running the PrivaMix CoC Edition. During operation, there can be only one PrivaMix CoC Edition operating on the network.

The other version of PrivaMix runs on a machine under the control of each data holder participating in the deduplication. This version of PrivaMix is termed *PrivaMix Shelter Edition*. Each data holder operates its own machine, where PrivaMix Shelter Edition runs on that machine. That machine also contains a copy of the data holder's client data that is the subject of the deduplication. Participants cannot share machines because all machines have to be operational on the network at the same time. Additionally, each copy of PrivaMix Shelter Edition has a pre-assigned unique serial number to facilitate isolated communication among network members. Therefore, no two machines operating over the network can have the same serial number. Each machine must have its own specific licensed copy of PrivaMix to insure a unique serial number.

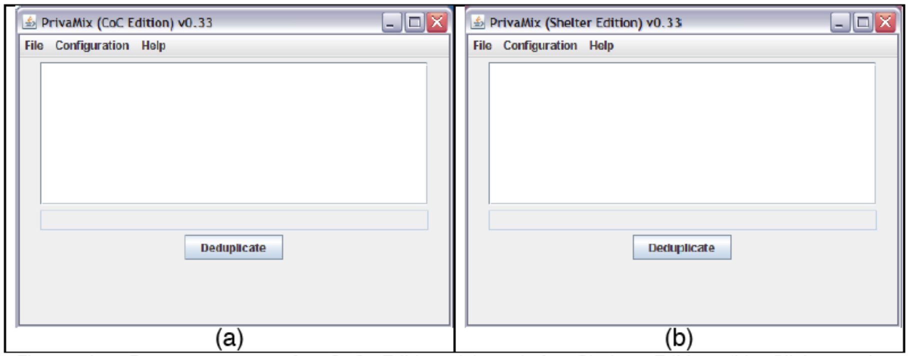
PrivaMix (all editions) runs under Windows, Mac OS, and Unix in basic machine configurations. The machine must have reliable access to the Internet during the time the program deduplicates.

Quick Start

If the network has been pre-configured, all the default settings will allow immediate operation. In this case, participants do the following:

1. *Data holders*: Copy your client data file to c:\data.csv. The file must have this name and be located in this directory.
2. *CoC*: Load the program and then click on the *Deduplicate* button. See Figure 1(a) below.
3. *Data holders*: Load the program and then click on the *Deduplicate* button. See Figure 1(b) below.
4. *CoC*: The results of the deduplication appear in the files c:\results1.csv and results2.csv when the button is renamed to *Exit*.

If your network or machine has not been pre-configured or the default settings need to be confirmed, checked, or modified, see the remaining sections of this writing for specifics on how to make changes and check files.



(a) (b)

Figure 1. Program screen for CoC Edition (a) and for Shelter Edition (b). Click on the "Deduplicate" button to run the program.

3. Shelter Edition

Below is a description of the basic menu commands in PrivaMix Shelter Edition. The order in which a command appears in this writing is organized around the program display –File, Configuration, and Help. Click on the *Deduplicate* button to execute the program.

Deduplicate

Click the *Deduplicate* button, which prominently appears on the main program screen to execute the program. If a problem occurs, use the menu commands to change the configuration to match the selected data file and network. Figure 2 shows the Deduplicate button as it appears in the program window.

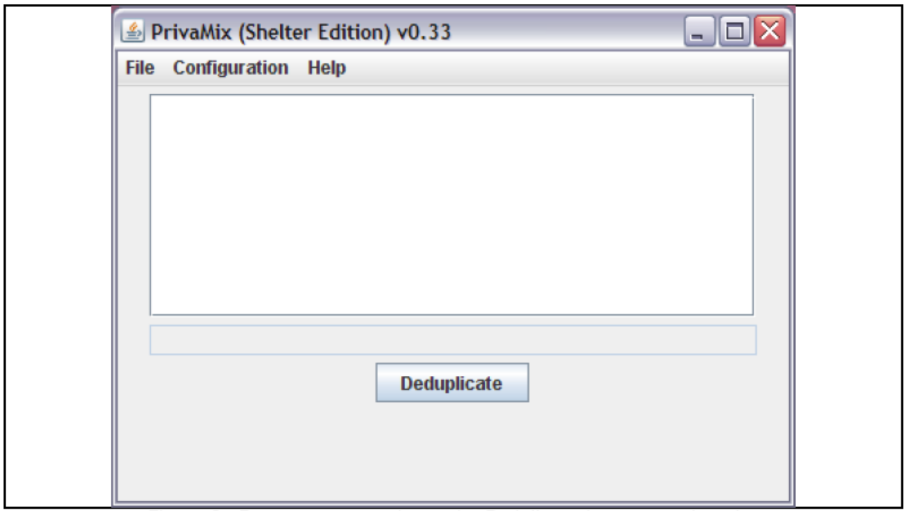


Figure 2. Program screen shows the Dedplicate button prominently.

File Menu

The *File* Menu has operations related to the file that contains the client information that is the subject of the deduplication. Figure 3 shows the available operations, which include *Select* and *Check*. These operations are described below. The File Menu also includes an *Exit* operation to terminate the program.

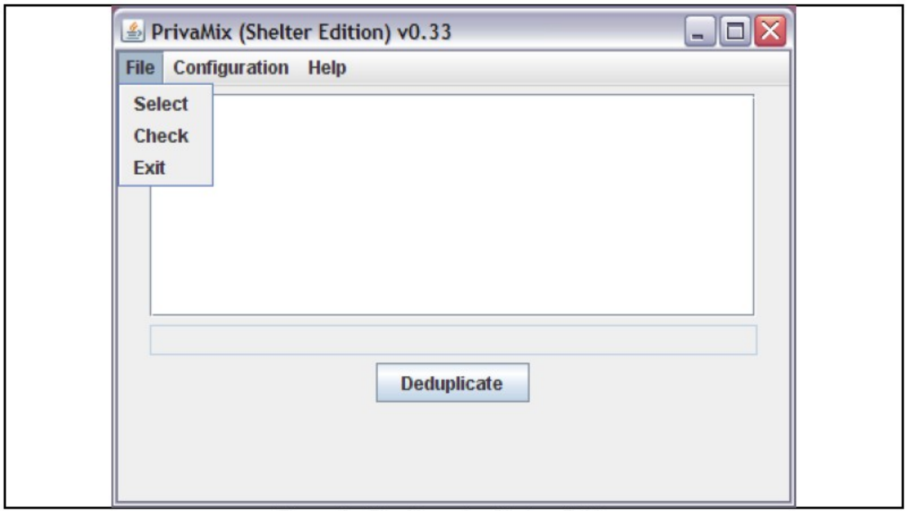


Figure 3. File Menu for Shelter Edition.

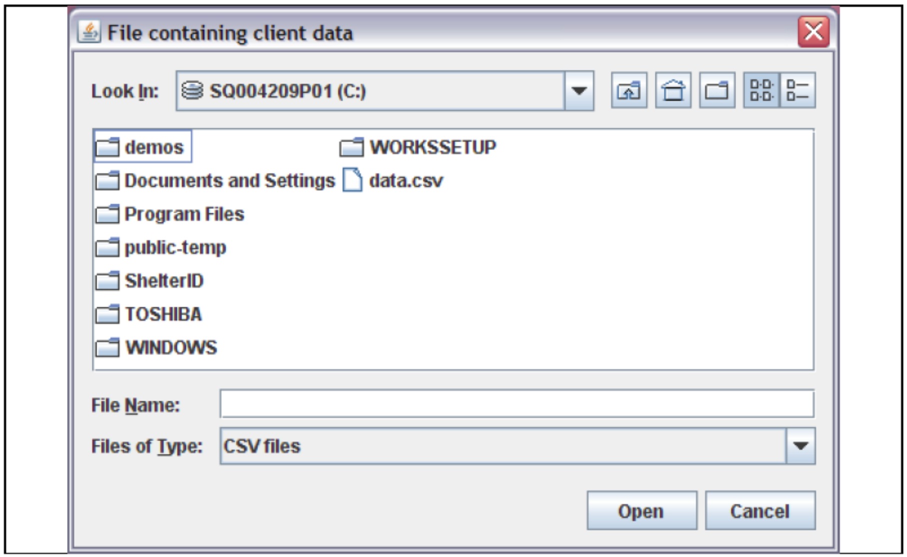
File-Select

Clicking on *File*, then *Select* displays a window for navigating through the file system in order to locate the file that contains the client information that is the subject of the deduplication. This file should be a CSV (comma-delimited file), whose filename ends in ".txt", ".TXT", ".csv", or ".CSV". Figure 4 shows an example of the navigation window that appears.

To see which data file is currently selected, use the Configuration-Current filename command.

To check the integrity of the format file to make sure it matches the program's current expectations, use the File-Check command.

To run the program, click the Deduplicate button.



File containing client data

Look In: SQ004209P01 (C:)

File Name:

Files of Type: CSV files

Figure 4. File-Select command for Shelter Edition pops up a navigation window.

File-Check

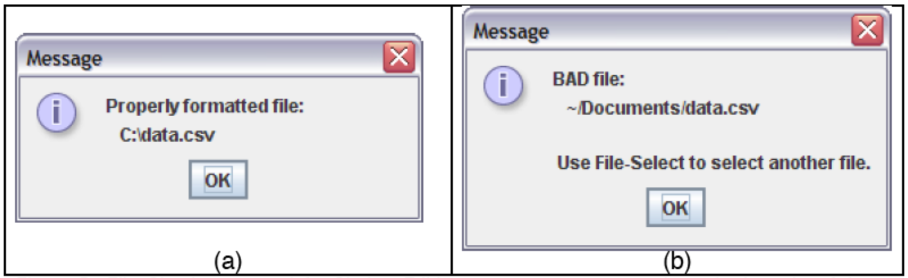
Clicking on *File*, then *Check* runs an integrity check on the data file currently selected and reports the result as good or bad. Figure 5 shows the good and bad result messages. The filename of the file that is checked appears in the pop-up window, as shown in Figure 5.

To see what data file is already selected, use the Configuration-Current filename command.

To change the data file to use, use the File-Select command.

To change the format expected of the file, use the Configuration-CSV format command.

To run the program, click the Deduplicate button.



Properly formatted file:
C:\data.csv

BAD file:
~/Documents/data.csv

Use File-Select to select another file.

Figure 5. File-Check command for Shelter Edition pops up a message about the integrity of the file as good (a) or bad (b).

Configuration Menu

The *Configuration* menu contains commands for altering default values used throughout the system. Figure 6 shows a list of available configuration commands.

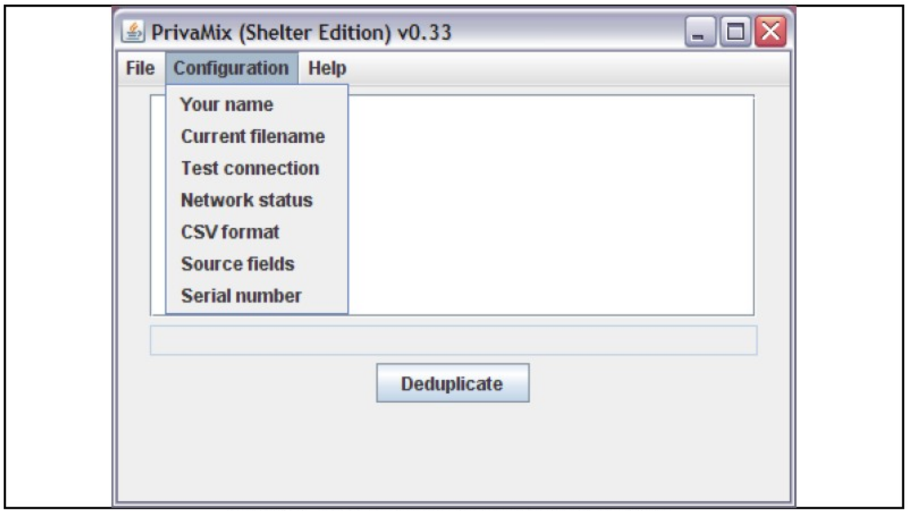


Figure 6. Configuration menu for Shelter Edition.

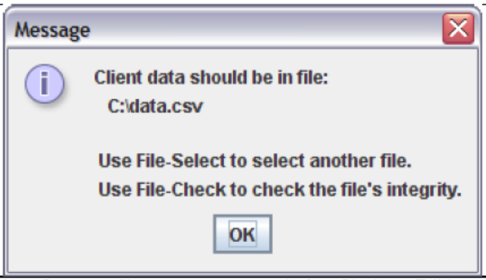
Configuration-Current filename

Clicking on *Configuration*, then *Current filename* displays the name of the file currently selected as the file containing the client information for deduplication. Figure 7 shows an example of the pop-up window.

To change the data file to use, use the File-Select command.

To change the format expected of the file, use the Configuration-CSV format command.

To run the program, click the Deduplicate button.



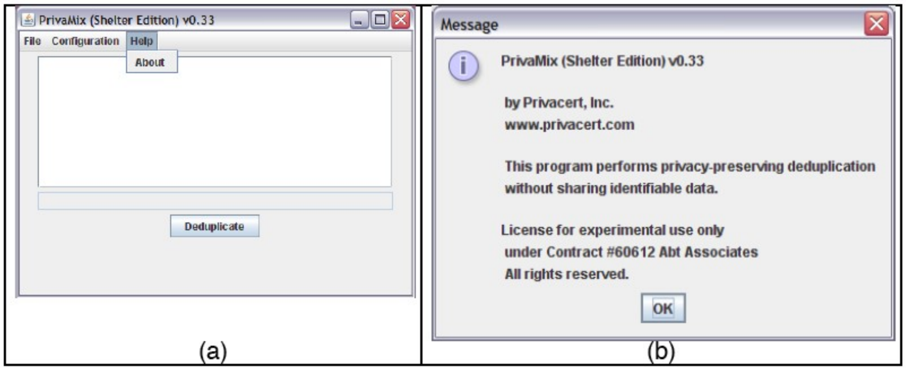
Client data should be in file:
C:\data.csv

Use File-Select to select another file.
Use File-Check to check the file's integrity.

Figure 7. Configuration-Current filename command for Shelter Edition pops up a message displaying the filename of the file currently selected for processing.

Help Menu and the Help-About Command

The *Help* menu contains the About command which displays version and license information for the copy of PrivaMix running. Figure 8 (a) shows the Help menu and Figure 8(b) shows the pop-up window that appears with the Help-About command.



PrivaMix (Shelter Edition) v0.33

by Privacert, Inc.
www.privacert.com

This program performs privacy-preserving deduplication without sharing identifiable data.

License for experimental use only
under Contract #60612 Abt Associates
All rights reserved.

Figure 8. Help menu (a) and the results of the Help-About command (b).

3. CoC Edition

Below is a description of the basic menu commands in PrivaMix CoC Edition. These commands are similar to those found in the Shelter Edition, mentioned previously. Click on the Dediplicate program to execute the program.

Deduplicate

Click the *Deduplicate* button, which prominently appears on the main program screen to execute the program. If a problem occurs, use the menu commands to change the configuration to match the selected data file and network. Figure 9 shows the Deduplicate button as it appears in the program window.

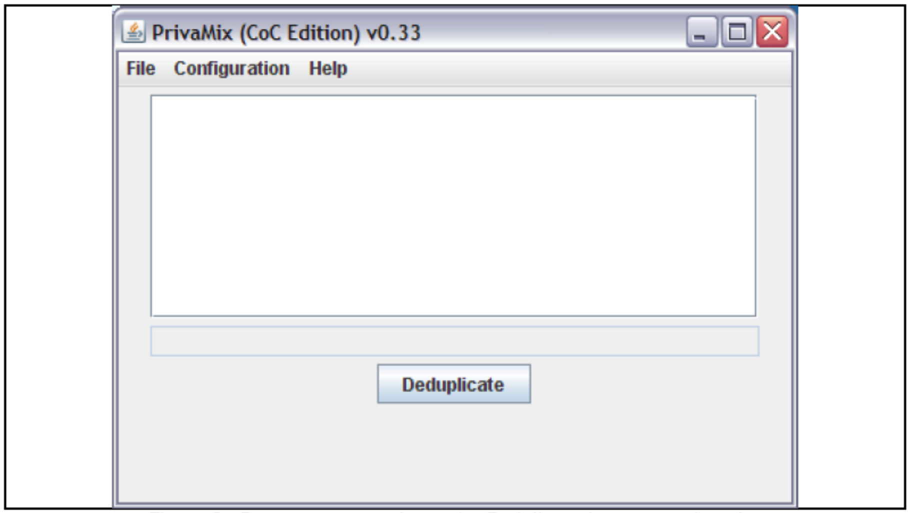


Figure 9. Program screen shows the Dedplicate button prominently.

File Menu (CoC Edition)

The *File* Menu has the *Save as* operation, which allows the user to provide the names of the files files that will store the results of the deduplication. The File Menu also includes an *Exit* operation to terminate the program. Figure 10 shows the available operations.

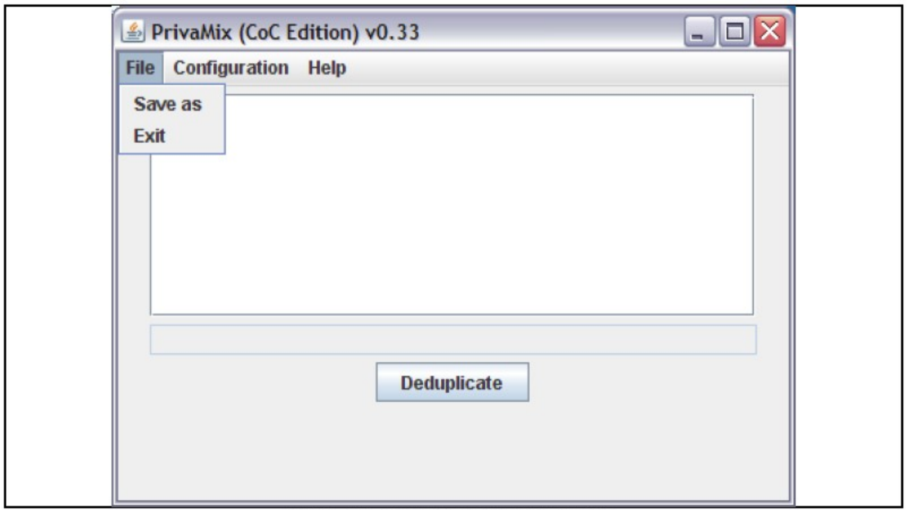


Figure 10. File Menu for CoC Edition.

Configuration Menu (CoC Edition)

The *Configuration* menu contains commands for altering default values used throughout the system. Figure 11 shows a list of available configuration commands.

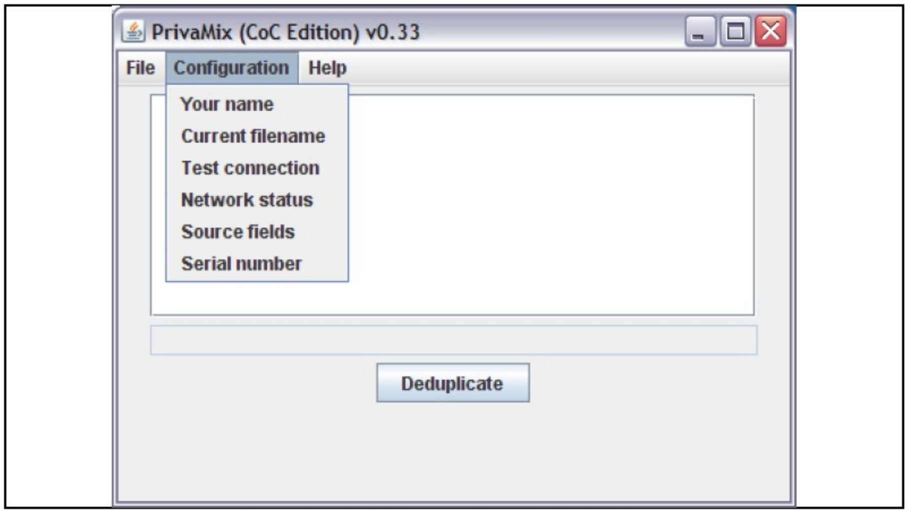
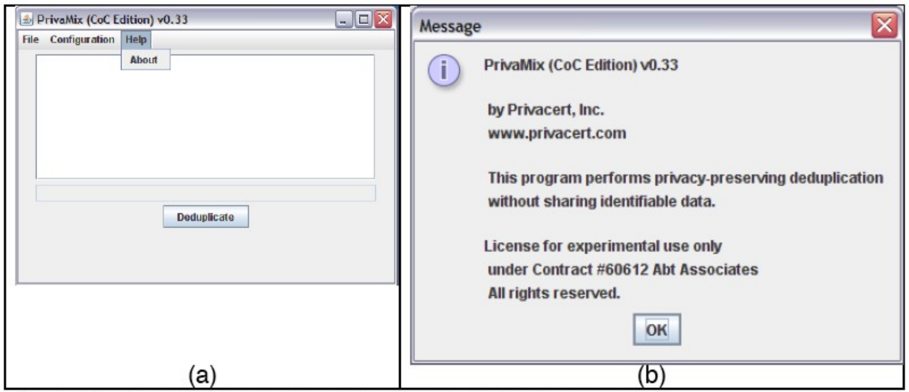


Figure 11. Configuration menu for CoC Edition.

Help Menu and the Help-About Command (CoC Edition)

The *Help* menu contains the *About* command which displays version and license information for the copy of PrivaMix running. Figure 12 (a) shows the Help menu and Figure 12(b) shows the pop-up window that appears with the Help-About command.



PrivaMix (CoC Edition) v0.33

by Privacert, Inc.
www.privacert.com

This program performs privacy-preserving deduplication without sharing identifiable data.

License for experimental use only
under Contract #60612 Abt Associates
All rights reserved.

(a) (b)

Figure 12. Help menu (a) and the results of the Help-About command (b).

Sweeney, L. *Demonstration of a Privacy-Preserving System that Performs an Unduplicated Accounting of Services across Homeless Programs*. U.S. Government Release October 2008.

Index