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.

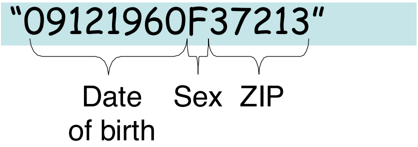


Figure 22. Example of making a UID by encoding {date of birth, gender, ZIP}.

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).

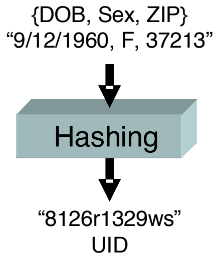


Figure 25. Example of making a UID by hashing {date of birth, gender, ZIP}.

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.

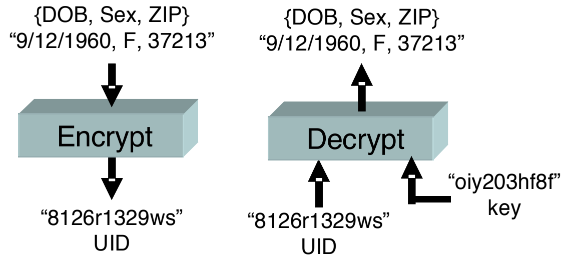


Figure 28. Example of making a UID by encrypting {date of birth, gender, ZIP}. With the key, the process is reversed to reveal the original source information.

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,

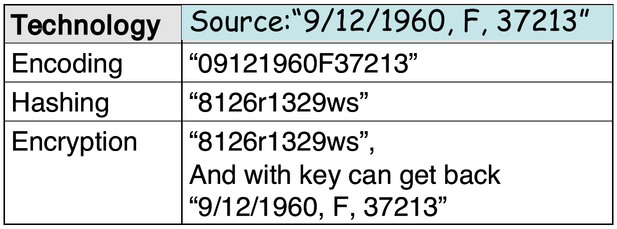
decrypt(*key*, "8126r1329ws") = "9/12/1960, F, 37213"

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.

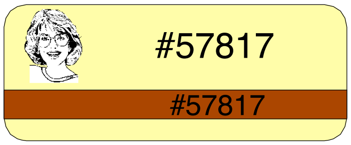


Figure 32. Depiction of a scan card with a serial number and photograph visible. The magnetic strip stores the serial number, but the serial number stored on the strip is not visible to the naked eye.

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.

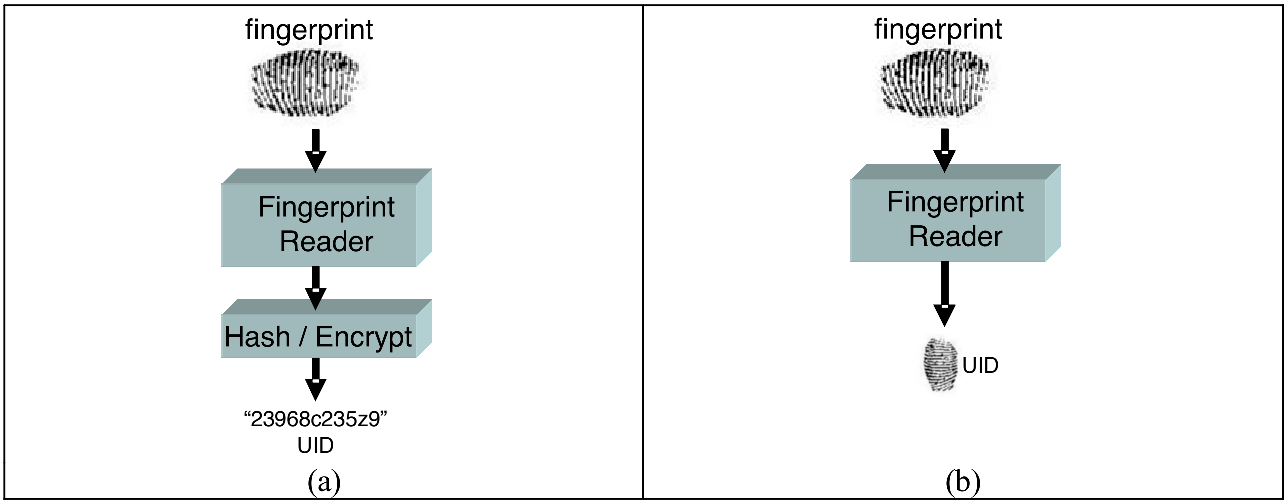


Figure 35. Fingerprint as source information to a hash or encryption function to generate a UID (a); or, used as the UID iteself (b).

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.

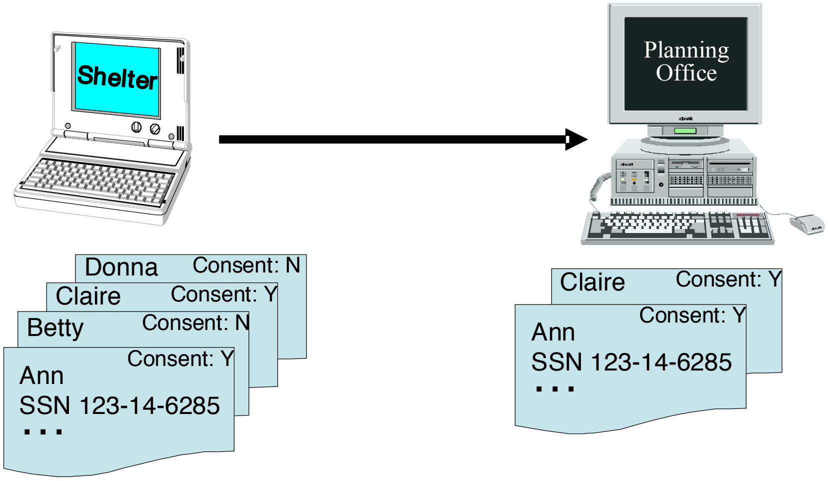


Figure 38. Consent used as the basis for deciding which Client information is forwarded to the Planning Office. Information provided to the Planning Office is explicitly identified by name and Social Security number.

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

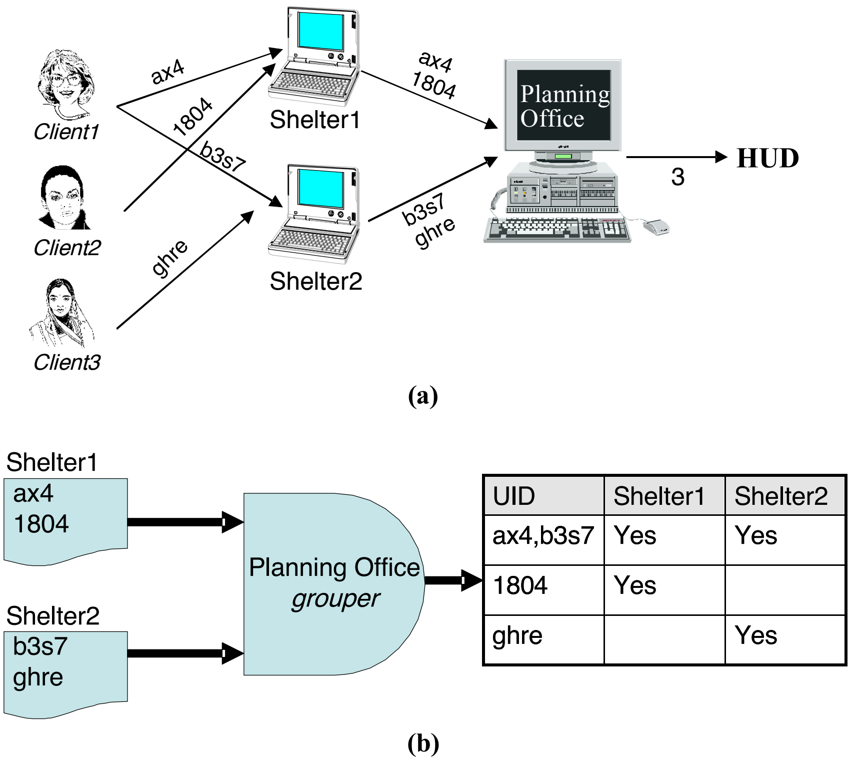
System Trust
Which parties are heavily trusted?

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 ≠ 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

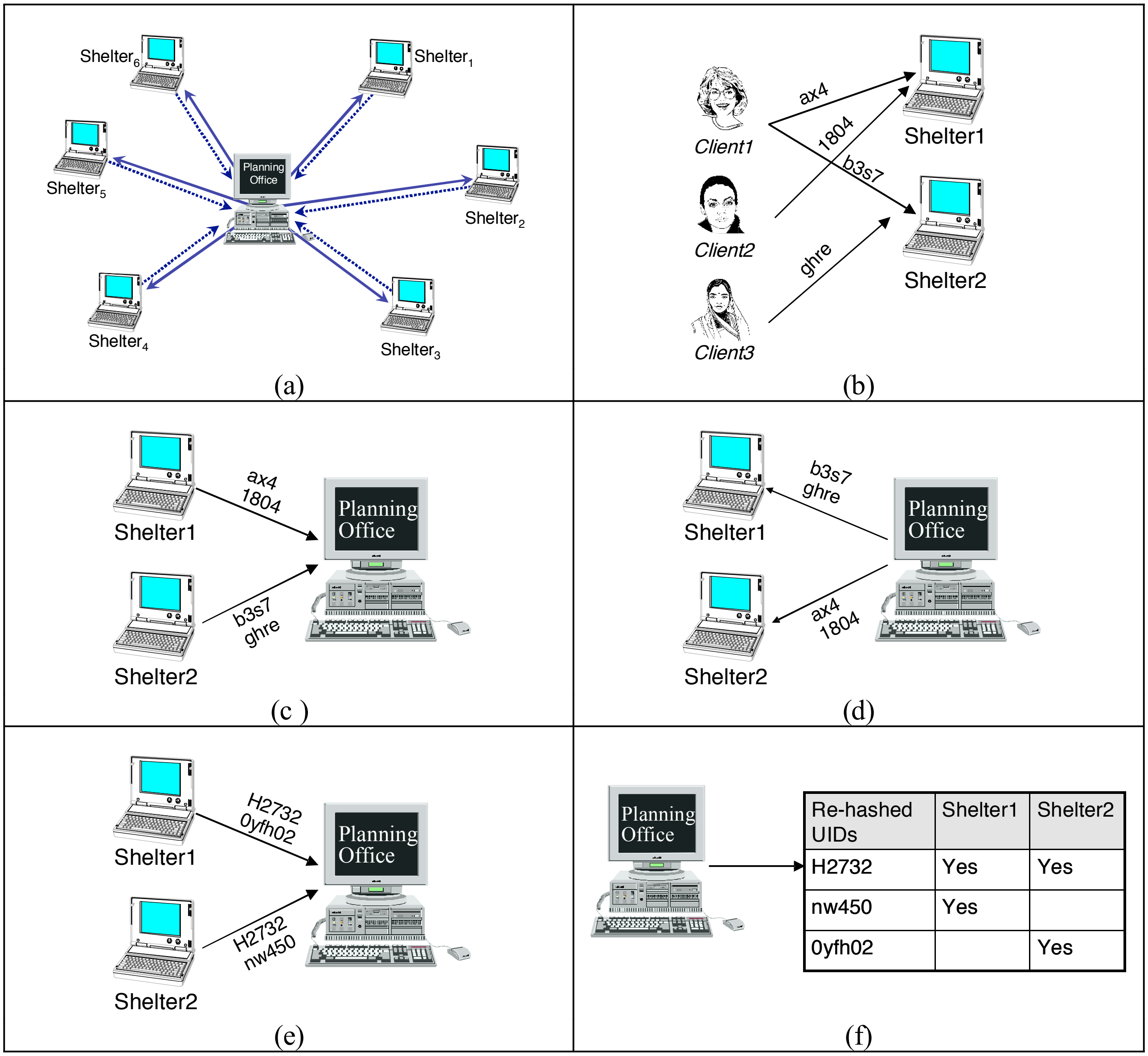
System Trust
Which parties are heavily trusted?

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 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.