Status	Letter Ballot
Date of Last Update	2020/01/16
Person Assigned	David Clunie mailto:dclunie@dclunie.com
Submitter Name	David Clunie mailto:dclunie@dclunie.com
Submission Date	2019/06/17

Correction Number CP-1955
Log Summary: De-identification and datetime stamps in UIDs
Name of Standard PS3.15
Rationale for Correction: UIDs may contain date and time stamps that may be related to visit date and hence an identity leak.
Correction Wording:

*Amend DICOM PS3.15 as follows (changes to existing text are bold and **underlined** for additions and **~~struckthrough~~** for removals):*

E.3.9 Retain UIDs Option

Though individuals do not have unique identifiers themselves, studies, series, instances and other entities in the DICOM model are assigned globally unique UIDs. Whilst these UIDs cannot be mapped directly to an individual out of context, given access to the original images, or to a database of the original images containing the UIDs, it would be possible to recover the individual's identity.

However, there are applications that require the ability to maintain an audit trail back to the original images and though there are other mechanisms they may not scale well or be reliably implemented. This Option is provided for use when it is judged that the risk of gaining access to the original information via the UIDs is small relative to the benefit of retaining them.

When this Option is specified in addition to an Application Level Confidentiality Profile, UIDs shall be retained, as described in ???.

Note

1. A UID of a DICOM entity is not the same as a unique identifier of an individual, such as would be proscribed by some privacy regulations.

2. UIDs are generated using a hierarchical scheme of "roots", which may be traceable by a knowledgeable person back to the original assignee of the root, typically the device manufacturer, but sometimes the organization using the device.

3. When evaluating the risk of matching UIDs with the original images or PACS database, one should consider that even if the UIDs are changed, the pixel data itself presents a similar risk. Specifically, the pixel data of the de-identified image can be matched against the pixel data of the original image. Such matching can be greatly accelerated by comparing pre-computed hash values of the pixel data. Removal of burned-in identification may change the pixel data but then matching against a sub-region of the pixel data is almost certainly possible (e.g., the central region of an image). Even addition of noise to an image is not sufficient to prevent re-identification since statistical matching techniques can be used. Ultimately, if any useable pixel data is retained during de-identification, then re-identification is nearly always possible if one has access to the original images. Ergo, replacement of UIDs should not give rise to a false confidence that the images have been more thoroughly de-identified than if the UIDs are retained.

4. Regardless of this option, implementers should take care not to remove UIDs that are structural and defined by the Standard as opposed to those that are instance-related. E.g., one would never remove or replace the SOP Class UID for de-identification purposes.

5. The Implementation Class UID (0002,0012) is not included in the list of UID attributes to be retained, since it is part of the File Meta Information (see ????), which is entirely replaced whenever a file is stored or modified during de-identification. See ???.

6. **UIDs may have been generated using a date/time stamp mechanism to assist in uniqueness generation. Hence if the original UIDs are not replaced, they may assist identity recovery. That said, the presence of an apparent date/time stamp in a UID does not indicate that it is an original UID, since replacement UIDs may contain new date/time stamps related to when the instance was de-identified. Also, UUIDs may be time-based rather than name-based or random-number-based, so use of the PS3.5 Section B.2 "UUID Derived UID" mechanism does not necessarily avoid this issue.**