Home Office

CAST Key Reference Hard Disk
Preparation procedure

Publication number 062/16

Andrew Barnes, Paul Farr, Jade James, Pat Mason

November 2016

Summary

The CAST Key Reference Hard Disk (CKRHD) consists of a conventional mechanical hard disk drive of a specific capacity; the complete user addressable storage capacity which is present on the disk is prepared with known data. The purpose of the CKRHD is to allow users to assess the extent to which an imaging process acquires an image of the user addressable area of a hard disk drive. More information can be found in CAST publication 063/16.

The CKRHD is intended to test the following key requirements of an imaging process (see ENFSI, 2015):

- A complete copy of the persistently stored user-addressable data on the evidence item, as presented at the time of examination by the disk controller using the Logical Block Address (LBA) scheme, shall be acquired.
- The acquired image shall replicate the structure, order and contents of the user-addressable storage on the evidence item at the time of creation of the image.

This document provides information about the preparation procedure used by CAST in the creation of each CKRHD over the period July 2015 – April 2016. The procedure complies with the principles which are detailed in CAST publication 063/16. Users may wish to refer to both documents in order to understand the scope and utility of the CKRHD, how it was created and how it may be applied to local method validation tasks. Competent users may use this procedure to prepare their own reference disks in the style of a CKRHD if desired.

Included within this document is a discussion about specific design decisions related to the CKRHD which were produced by CAST. The equipment used, and the key procedural steps in preparing the reference disks, is noted. The functionality of the bespoke software developed by CAST is addressed and consideration given to the character of the data which is generated.

Contents

1. Design considerations

Whilst determining an appropriate design for the CKRHD, CAST consulted with the lead forces and expert network as established by Deputy Chief Constable Nicholas Baker, the lead for digital forensics on the National Police Chiefs' Council (NPCC). The decisions detailed below were taken as a result of these consultations and with the agreement of the expert network. The guiding principle was to ensure that the CKRHD is as simple and generally applicable as possible.

Note that certain variations in local user requirements may be incompatible with these design decisions and local users must consider the applicability of the CKRHD within their own circumstances. Local users creating their own reference disks may choose an alternative design, although consideration must be given to the implications on the preparation of the reference disk and its subsequent use and utility.

1.1. Interface

The definition of a conventional hard disk drive in CAST publication 063/16 states that it communicates via an ATA interface. There are a range of physical interfaces which may be utilised by a device in the ATA family. The CKRHD is based upon a SATA variant since it is a ubiquitous consumer interface which is commonly encountered in case work. Local users must consider the risks presented by devices which use alternative physical interfaces and mitigate them appropriately.

1.2. User addressable data

Imaging is defined in CAST publication 063/16 as the capture of the persistently stored user-addressable data on the hard disk as presented to the host by the disk controller using the Logical Block Address (LBA) scheme. The ATA protocol allows for the presence of data which is obscured by the use of a Host Protected Area (HPA) and/or a Device Configuration Overlay (DCO) that may still be accessed by a knowledgeable user. Due to variations in local requirements relating to the capture of obscured data, and to maintain the simplicity and wider applicability of the CKRHD, HPA and DCO are not included within the scope of the user-addressable data. Therefore, no such obscured regions are included on the CKRHD. However, suitable obscured areas could be added to a standard CKRHD to match local requirements.

Data stored in service areas, firmware, or temporary caches are not generally available to typical users and as such are not within the scope of user-addressable data. Therefore, the data stored in these regions is not controlled or standardised and the CKRHD cannot be used to assess the extent to which an imaging process acquires data stored in these areas.

1.3. Disk capacity

The CKRHD features a small storage capacity of 80GB. Whilst larger capacity disks may stress the imaging process to a greater extent (for example, overheating of components may lead to inefficiency or failure of the process), the advantages of using a small disk in terms of minimising the time and storage overheads required to process the disk are compelling. The risks presented by larger capacity disks should be assessed and managed locally.

1.4. Data creation

The data content on the CKRHD must precisely match the full extent of the user-addressable storage capacity on the reference disk, and each sector of data present on the disk should feature unique, uncorrelated content (see CAST publication 063/16). Bespoke software based upon a standard, published pseudorandom number generator (Press and others, 2002) was created to conveniently generate a controlled data stream of a precise length which avoids sparse or repeated content (see section 4.1). The use of a fixed seed value within this software ensures that the data generated is entirely predictable and thus may be easily inspected for conformity.

1.5. Data assurance

The common technique by which users assess the consistency of the image obtained against the data on the disk is to compare summary identifiers such as the MD5 or SHA1 hash values of the data (Genoe, 2013). Providing users with the MD5 and the SHA1 hash value of the known data is therefore a sensible benchmark for the user to assess whether they have successfully imaged the reference disk; if they obtain the same hash values then it is reasonable to assume that they have acquired an accurate image of the disk. However, fixed-length hash functions such as MD5/SHA1 have, by definition, a limited key space and this should be considered when designing and risk assessing a local imaging procedure.

1.5.1. Limited key space

A hash function which generates a fixed length output can only produce a finite number of values. The number of possible inputs is infinite; therefore, some files will by definition hash to the same value. This is known as a collision. As a consequence it is possible that errors which occur in reading the data or writing it to a file could produce an output with the same hash value as the original, uncorrupted data. In reality, this likelihood is very low[1] as is detailed below.

Assuming an even distribution of outputs from the hash function, whereby each hash value is equally likely to occur, the probability of two different data streams accidentally producing the same hash value is as follows:

- MD5, 128 bit key, probability of accidental collision ~ 1 in 10^{38}
- SHA1, 160 bit key, probability of accidental collision ~ 1 in 10^{48}

Furthermore, assuming the MD5 and SHA1 functions are independent such that a pair of data streams which happen to collide in MD5 are no more likely than usual to collide in SHA1, the probability of two different data streams accidentally producing the same hash value for both MD5 and SHA1 is lower still:

- Probability of both MD5 and SHA1 accidentally colliding ~ 1 in 10^{86}

Both the MD5 and SHA1 hash values of the user-addressable data content stored on the CKRHD are pre-calculated and noted in the documentation which accompanies the disk.

[1] For comparison, the chance of winning the jackpot in the National Lottery is ~1 in 10^{7}. The chance of an accidental MD5 collision is approximately equivalent to winning the National Lottery jackpot five times in a row. An accidental SHA1 collision would be equivalent to the chances of winning the jackpot for seven weeks in a row.

Users of the CKRHD can then decide whether to use MD5, SHA1, or both when seeking assurance that their imaging process has correctly acquired the image of the CKRHD.

Further assurance can be gained by direct inspection and comparison of the data. The possibility of undetected corruption occurring in the disk preparation process was mitigated by directly inspecting samples of the data and by carrying out an automated byte-by-byte verification of the full extent of the data read from the disk against the data generated from the algorithm.

2. Equipment

2.1. Disk generating PC

The disk generating PC fulfilled the criteria outlined in CAST publication 063/16 regarding the selection of equipment. It was the machine used to characterise the reference disk, to create files containing known data, to characterise the known data and to put this data onto the reference disk. It was also used to verify the data on the prepared reference disk on a byte-by-byte basis.

- Model: Dell Precision T1700
- BIOS: A10
- Processor: Intel Core i7-4790
- Memory: 16 GiB DDR3 SDRAM
- OS: Ubuntu 14.04 LTS (64-bit)
  - Kernel: 3.16.0-30
  - hdparm: 9.43
  - GNU coreutils (dd, md5sum, sha1sum): 8.21
  - grep: 2.16
  - dmesg: 2.20.1
  - xxd: 1.10
- Disk image generator
  - Bespoke software utilising a L'Ecuyer number generator (see 4.1.1)
- Disk image checker
  - Bespoke software utilising a L'Ecuyer number generator (see 4.1.2)

2.2. Disk imaging PC

The disk imaging PC was arbitrarily chosen to be a Windows 7 build. This was used to verify that the CKRHD could be imaged as expected and to calculate the MD5 and SHA1 values of the images.

- Model: Dell Optiplex 7010
- BIOS: A08
- Processor: Intel Core i7-3770
- Memory: 16GiB DDR3 SDRAM
- OS: Windows 7 Ultimate SP1 (64-bit)
- Imaging software: FTK Imager 3.4.0.1
- Hardware write-blocker: Tableau T35es Ultrablock II (connected via FireWire IEEE1394)

3. Preparation procedure

The preparation procedure for the CKRHD involves the characterisation of the device which is to act as the reference disk, the generation and characterisation of known data to be placed on the disk, and the transfer of that data across to the disk. The procedure is completed by undertaking quality checks of the prepared disk and then completing a certificate of assurance.

3.1. Device characterisation

The initial stage of disk preparation establishes a baseline ground truth for the full extent of the user-addressable storage capacity as presented by the disk controller. This information will enable the generation of an appropriate quantity of known data to place on the CKRHD in subsequent stages.

a) Label the CKRHD using the format '*CKRHD_XXX*' where *XXX* is an index number which uniquely refers to that particular disk.

b) Create a record for the CKRHD in the reference disk database.

c) Visually inspect the disk. Record the serial number, manufacturer, model, sector size, number of sectors, interface and date of manufacture and photograph the disk label.

d) With the disk generating PC switched off, connect the CKRHD via SATA directly to the motherboard. This minimises the chance of accidental electrical damage or injury, whilst the chance of successful registration of the CKRHD on the disk generating PC is maximised.

e) Switch on the disk generating PC and allow it to complete its boot sequence.

f) Create a directory within the data area of the disk generating PC in which to record an audit trail for the preparation of the reference disk.

g) Establish the device name (for example, `/dev/sdb`) assigned to the disk by the operating system by querying the kernel message buffer using the following command:

```
dmesg | grep sd[a-z]
```

The reference disk is identified by matching the reported output to the disk specification. All further interaction with the disk in the following procedure uses the upper case character '`X`' as a proxy for the appropriate device name; **this upper case '`X`' should be replaced with the appropriate lower case alphabetic character**.

Note that the kernel message buffer includes the number of sectors which are reported by the disk. For audit purposes, create a text document to record this information by navigating to the directory for the disk and entering the following command:

```
dmesg | grep sdX > dmesg_output.txt
```

h) Confirm that DCO is not masking any sectors by entering the following command:

   ```
   sudo hdparm --dco-identify /dev/sdX
   ```

   and compare the reported number of sectors against the number established in the preceding step. For audit purposes, create a file confirming this information as follows:

   ```
   sudo hdparm --dco-identify /dev/sdX > dco_identify.txt
   ```

i) Confirm that HPA is not masking any sectors by entering the following command:

   ```
   sudo hdparm -N /dev/sdX
   ```

   and compare the reported number of sectors against the number established in the preceding steps. For audit purposes, create a file confirming this information as follows:

   ```
   sudo hdparm -N /dev/sdX > hpa_output.txt
   ```

3.2. Data generation

The *Disk Image Generator* program creates a stream of data which is written to a file whose size matches the capacity of the reference disk to be created.

a) From within the folder containing the *Disk Image Generator* software, create the known data file:

   ```
   ./DiskImageGenerator -n<sectornumbers> -f<filepath>
   ```

   where `<sectornumbers>` is the desired number of sectors to generate, as established during the characterisation of the reference disk, and `<filepath>` is the desired location of the generated file on the data drive partition.

3.3. Data characterisation

The generated data stream is characterised by calculating two deterministic summary values which identify the data stream.

a) Calculate and record the MD5 and SHA1 hash values of the generated data file as follows:

   ```
   md5sum <filename> > hashvalues.txt ;
   sha1sum <filename> >> hashvalues.txt
   ```

   where `<filename>` refers to the generated data file and `hashvalues.txt` is the output file containing the hash values. The use of "`>>`" ensures that the SHA1 hash value is appended (rather than overwritten) to the same file as the MD5 hash value.

3.4. Data transfer

The full extent of the generated and characterised data stream is transferred to the CKRHD.

a) Use the `dd` command to write the known data stream file to the reference disk as follows:

   ```
   sudo dd if=<filename> of=/dev/sdX bs=1M
   ```

   where `<filename>` is the name of the known data file and `/dev/sdX` is the device name assigned by the host operating system to the CKRHD. The block size can be adjusted if necessary to improve performance.

b) Record the metadata associated with the `dd` command, including time taken, records in and out and number of bytes transferred.

3.5. Disk assurance

Quality assurance checks performed on the prepared CKRHD prior to release include a combination of explicit byte-by-byte comparison of the data contents against the predetermined result, an imaging test, summary identifier comparison between the imaged and the generated data and direct manual inspection of the data present on the disk. Anomalous results will indicate an issue which should be resolved prior to release of the CKRHD.

a) From the directory created for the reference disk in step 3.1f), run *Disk Image Checker*:
```
sudo <path_to_Disk_Image_Checker> -f/dev/sdX
```
Examine the logfile generated by the checker program to confirm that no mismatches were identified.

b) Following successful completion of the *Disk Image Checker*, power down the disk generating PC and disconnect the CKRHD. Further disk assurance checks are completed using the disk imaging PC.

c) Attach the CKRHD to the disk imaging PC via a hardware write-blocker to minimise the risk of unwanted changes being made to the CKRHD.

d) Acquire a physical image of the CKRHD in *FTK Imager* using the following settings:
raw image file
fragment size 0 (do not fragment)
verification – selected
pre-calculate – unselected
create a directory listing– unselected

e) Confirm that the MD5 and SHA1 hash values recorded by the imaging process match those obtained during the known data file generation process.

3.6. Completing the certificate of assurance

This step consists of a review of the information generated during the preparation and assurance of the reference disk. Access to both the disk generating and disk imaging PC will be required. The review is recorded in the formal certificate to be distributed to each local user along with the reference disk. The certificate acts as a formal record of the quality assurance checks which were undertaken for the reference disk.

a) Confirm the disk name (*CKRHD-XXX*) and the manufacturer's serial number and record the information on the certificate.

b) Transcribe the LBA extent from the log file created when reviewing the kernel message buffer. Cross-check this value against the log files created when HPA and DCO were inspected.

c) Confirm the absence of HPA and DCO from the log file on the disk generating PC.

d) Transcribe the *Disk Image Generator* version from the log files on the disk generating PC.

e) Transcribe the "write time to disk" from the `dd` log files on the disk generating PC. This time should be similar to the time taken to complete the *Disk Image Checker* program (see step 3.6i) below) and the time taken to image the CKRHD (see step 3.6l) below). Any significant discrepancies between the times taken to write, check and image could indicate a problem or inconsistency with the reference disk which should be investigated further prior to release.

f) Record the name of the individual who created the CKRHD in the "written by" field.

g) Record the date that the CKRHD was created in the “write date” field.

h) Enter the identifying asset number of the disk generating PC in the *Disk Image Checker* machine field, together with the character ‘*u*’ to indicate the *Ubuntu* operating system.

i) Transcribe the version of the *Disk Image Checker* program from the log file, along with the time taken to complete the check. Confirm that no mismatches were reported and record the result.

j) Enter the asset number of the disk imaging PC in the imaging verification machine field, together with a ‘*w*’ to indicate the *Windows* operating system.

k) Transcribe in to the “Tool” field the version of *FTK Imager* used in the disk imaging PC.

l) Calculate and transcribe the time taken to image the CKRHD from the *FTK Imager* log file.

m) Record the LBA extent from the *FTK Imager* log file. Cross-check against the LBA extent recorded earlier on the form.

n) Transcribe the final 16 bytes of the *Disk Image Generator* data stream from the output of the following command on the disk generating PC:

```
dd if=<filename> skip=<lba_extent-1> bs=512 | xxd
```

where `<filename>` is replaced with the generated data file details. The `skip` and `bs` switches filter the output to only include the final sector (512 bytes), and `xxd` renders the output into a human-readable format.

o) Obtain the final 16 bytes on the disk by examining the image produced by *FTK Imager*.

p) Transcribe the MD5 hash value and the SHA1 hash value from the `hashvalues.txt` log file on the disk generating PC and from the *FTK Imager* log file on the disk imaging PC.

q) The individual who transcribes the entries signs the form as the verifying officer.

r) A second individual (the authorising officer) then reviews and signs the completed certificate, ensuring that the details are completed accurately.

4. Bespoke software

4.1. Software description

Two separate programs were developed in order to assist with the preparation of the CKRHD. The first program can create a file containing a controlled stream of data to place on the CKRHD. The second program can read in a file and check whether the data conforms to the expected output from the first program.

The source code for the software is available from the GOV.UK website[2]. It is written in C++ and once compiled requires no external libraries, allowing for portability across platforms. It is intended to be run from a command line interface.

4.1.1. Disk Image Generator

The disk image generator creates a file of a specific size which is filled with a controlled and reproducible sequence of data. The program is passed the file name and the intended size (in 512-byte sectors) on the command line, then opens the output file for writing. It iteratively generates sector-long sequences of data using a pseudorandom number generator seeded with a fixed initial value. Each sector is written to the file in turn before finally closing the output file.

The data generator algorithm is adapted from the L'Ecuyer number generator found in Press and others (2002). It was selected as an efficient and repeatable means of generating a long[3] sequence of data which exhibits little autocorrelation. This characteristic reduces the likelihood of repeated or identical sectors being present on the CKRHD.

4.1.2. Disk Image Checker

The disk image checker reads a file and checks whether it is filled with the expected sequence of data. It is intended to check that a data stream is exactly the same as would have been generated by the disk image generator program, and to identify the location within the data stream of any instances where the predetermined sequence and the file data do not match.

4.2. Software output assurance

To guarantee detection of imaging errors which occur at a sector level, the data generated by the bespoke software should not feature repeated or identical sectors. However, it is possible that the pseudorandom number generator function will repeat some extended sequences of bytes when creating a large quantity of data. Although the chance of identical sectors being

[2] See https://www.gov.uk/government/publications/cast-key-reference-hard-disk-preparation-procedure

[3] The algorithm is based on the difference between two periodic pseudorandom number sequences. The individual sequences possess periods of over 10^9 iterations before they begin to repeat. The difference between these two sequences has a theoretical period of over 10^{18} iterations before repetition, which is greater than that necessary to generate 1TB (~10^{12} bytes) of data.

generated is extremely unlikely[4], tests have been performed to assess whether any identical sectors occur on the CKRHD.

4.2.1. Test for the presence of identical sectors

A simple program that checks each sector, byte for byte, against each other sector would be computationally inefficient. By utilising an iterative sorting algorithm the problem complexity can be reduced; for an ordered set of sectors, collisions can be detected by comparing the current sector with the next sector in the list. The complexity can be further reduced by limiting the search for collisions to the first eight bytes of each sector; the consequent reduction in the data volume makes it possible to perform large sections of the data sorting in physical memory.

The pseudorandom number generating function was put into a test harness that generated data sector by sector with the same algorithm and seed value as used in the software described in 4.1.1 and 4.1.2. For each sector, the first eight bytes were appended to a file based upon the value of the first byte of the sector. When all of the data had been generated, each of the files was opened, sorted and compared.

If any of the byte sequences are found to be identical, an image generated by the standard disk image generator software can be searched for the detected collision byte sequences. The remaining data in these sectors can be manually compared to establish whether or not the remaining data content of the sectors is identical.

The test harness was run to generate 80GB of data, matching the capacity of the CKRHD. No collisions were detected between the first eight bytes of each sector of the data generated.

Based on this test, CAST can state that the standard disk image that has been shipped by CAST does not have any repeated sectors.

[4] Since each sector is an ordered list of 512 bytes, and each byte may have 256 possible values, the number of potential different arrangements of data within a sector is huge ($256^{512} \approx 10^{1233}$ permutations). It is highly unlikely that, by pure chance, any two given sectors will be the same. Assuming the data generator is perfectly random such that each possible output from the generator is always equally likely (and therefore that the data content of each individual sector is independent of any preceding or successive sectors), the theoretical cumulative chance of identical sectors occurring within 80GB of data is exceptionally close to zero; the cumulative chance of sector ‘n’ being identical to any of the preceding sectors can be approximated to p(match) $\approx 1 - e^{-n^2/2(256^{512})}$

5. References and acknowledgements

5.1. References

CAST (2016) *Fundamental principles for preparing the CAST Key Reference Hard Disk*. Publication 063/16.

ENFSI (2015) *Best practice manual for the forensic examination of digital technology*. European Network of Forensic Science Institutes, Forensic Information Technology Working Group, Version 01.

Genoe, R. (2013) *Managing a digital investigation unit – a handbook for senior law enforcement officers*. University College Dublin Centre for Cybersecurity and Cybercrime Investigations, page 19.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P. (2002) *Numerical recipes in C++: the art of scientific computing*. Cambridge University Press, 2nd edition, pages 279-290.

5.2. Acknowledgements

Grateful thanks to the Information Security Research Group at the University of South Wales for peer reviewing the technical method.

6. Glossary

ATA	Advanced Technology Attachment. An interface standard featuring a common command set.
CAST	The Home Office Centre for Applied Science and Technology.
CKRHD	CAST Key Reference Hard Disk.
DCO	Device Configuration Overlay.
ENFSI	European Network of Forensic Science Institutes.
HPA	Host Protected Area.
LBA	Logical Block Address.
MD5	Message Digest 5, a cryptographic hashing function.
NPCC	National Police Chiefs' Council.
SATA	Serial ATA, an interface standard for the connection of storage devices.
SHA1	Secure Hashing Algorithm 1, a cryptographic hashing function.

ISBN: 978-1-78655-131-3