IEEE SA
STANDARDS ASSOCIATION

IEEE Standard for Sanitizing Storage

STANDARDS

IEEE Computer Society

Developed by the
Cybersecurity and Privacy Standards Committee

IEEE Std 2883™-2022

IEEE

IEEE Standard for Sanitizing Storage

Developed by the

Cybersecurity and Privacy Standards Committee
of the
IEEE Computer Society

Approved 16 June 2022

IEEE SA Standards Board

Abstract: Methods for sanitizing logical storage and physical storage, as well as for providing technology-specific requirements and guidance for the elimination of recorded data, are specified in this standard.

Keywords: clear, crypto erase, cryptographic erase, crypto scramble, data destruct, data removal, destroy, IEEE 2883™, media sanitization, purge, sanitization, sanitize, security

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

PDF: ISBN 978-1-5044-8856-3 STD25526
Print: ISBN 978-1-5044-8857-0 STDPD25526

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (https://standards.ieee.org/ipr/disclaimers.html), appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents."

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE standards documents are supplied "AS IS" and "WITH ALL FAULTS."

Use of an IEEE standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents**.

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and Standards Coordinating Committees are not able to provide an instant response to comments, or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the IEEE SA myProject system.[1] An IEEE Account is needed to access the application.

Comments on standards should be submitted using the Contact Us form.[2]

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

[1] Available at: https://development.standards.ieee.org/myproject-web/public/view.html#landing.
[2] Available at: https://standards.ieee.org/content/ieee-standards/en/about/contact/index.html.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; https://www.copyright.com/. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit IEEE Xplore or contact IEEE.[3] For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE SA Website.[4] Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in IEEE Xplore. Users are encouraged to periodically check for errata.

[3] Available at: https://ieeexplore.ieee.org/browse/standards/collection/ieee.
[4] Available at: https://standards.ieee.org/standard/index.html.

Patents

IEEE Standards are developed in compliance with the IEEE SA Patent Policy.[5]

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at https://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. IEEE Standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

[5] Available at: https://standards.ieee.org/about/sasb/patcom/materials.html.

Participants

At the time this IEEE standard was completed, the Security in Storage Working Group had the following membership:

Jim Hatfield, *Chair*
Eric Hibbard, *Vice Chair*

Richard Austin
Moshin Awan
Sridhar Balasubramanian
David Black
Joseph Chen
Tim Chevalier
Tim Courtney
Anthony Duran
Monty Forehand
John Geldman
JonMichael Hands
Michael Harstrick
Chris Hillier
Walt Hubis
Glen Jaquette
B.J. Lang
Chandra Nelogal
Jamie Pocas
Anna Polubaryeva
Thomas Rivera
Yoni Shternhell
Curtis Stevens
Paul Suhler
Gary Sutphin
Danny Ybarra
Jemmee Yung

The following members of the individual Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Robert Aiello
Johann Amsenga
Philippe Astier
Richard Austin
Ted Bardusch
David Black
Koti Reddy Butukuri
Peter Capelli
Juan Carreon
Joseph Chen
Roger Cummings
Ronald Dean
Andrew Fieldsend
Thomas Friend
John Geldman
Jim Hatfield
Mariana Hentea
Jonathan Herrera
Shui Heung
Eric Hibbard
Chris Hillier
Werner Hoelzl
Richard Jessop
Piotr Karocki
N. Kishor Narang
Henry Newman
Jamie Pocas
R. K. Rannow
Lakshman Raut
Thomas Rivera
Scott Robertson
Bartien Sayogo
Walter Struppler
Paul Suhler
Dmitri Varsanofiev
John Vergis
Forrest Wright
Oren Yuen

When the IEEE SA Standards Board approved this standard on 16 June 2022, it had the following membership:

David J. Law, *Chair*
Ted Burse, *Vice Chair*
Gary Hoffman, *Past Chair*
Konstantinos Karachalios, *Secretary*

Edward A. Addy
Ramy Ahmed Fathy
J. Travis Griffith
Guido R. Hiertz
Yousef Kimiagar
Joseph L. Koepfinger*
Thomas Koshy
John D. Kulick
Johnny Daozhuang Lin
Kevin Lu
Daleep C. Mohla
Andrew Myles
Damir Novosel
Annette D. Reilly
Robby Robson
Jon Walter Rosdahl
Mark Siira
Dorothy V. Stanley
Lei Wang
F. Keith Waters
Karl Weber
Sha Wei
Philip B. Winston
Daidi Zhong

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 2883-2022, IEEE Standard for Sanitizing Storage.

Various data types are recorded on a range of data storage technologies. When these systems or their storage media are repurposed or retired from use, access to the recorded data often needs to be eliminated (sanitized) to avoid unauthorized access to the data. Depending on the storage technology, specific methods can be employed to help ensure that the data are either eliminated or the logical storage and physical storage associated with the data devices/storage media are disposed of properly.

The stakeholders for this standard include all consumers of data storage technologies, especially those that store sensitive or high-value data, and the vendors that manufacture, maintain, and support these technologies. Additionally, regulators and other standards development organizations can leverage the contents of this standard.

Contents

List of Figures

List of Tables

IEEE Standard for Sanitizing Storage

1. Overview

1.1 Scope

This standard specifies methods for sanitizing logical storage and physical storage, as well as for providing technology-specific requirements and guidance for the elimination of recorded data.

1.2 Using this standard

The reader can use this standard to do the following:

- become familiar with the principles of storage *sanitization* in Clause 5;
- find the details of the relevant *sanitization* techniques in Clause 6;
- review options and issues for verification of *sanitization* outcomes in Clause 7; and
- find a description of *sanitizing* the specific *storage device* type of interest in Clause 8.

Readers can find conformance details in 5.3, 5.4, and Clause 8.

1.3 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).[1,2]

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

2. Normative references

The following referenced documents are indispensable for the application of this standard (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

No normative references are cited in this standard.

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this standard, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.[3]

addressable: A characteristic of data, locations, or physical *storage media* in a storage device indicating an ability to be read or written through a *host interface*.

clear: Sanitize using logical techniques on *user data* on all *addressable* storage locations for protection against simple noninvasive data recovery techniques using the same *host interface* available to the user.

cryptographic erase: Method of sanitization in which the encryption key for the encrypted *target data* is *sanitized*, making recovery of the decrypted *target data* infeasible using state-of-the-art laboratory techniques.

degauss: Render magnetically stored data unreadable by applying a strong magnetic field to *storage media* with an organizationally approved field strength.

[1] The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.
[2] The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is only used in statements of fact.
[3] *IEEE Standards Dictionary Online* is available at: http://dictionary.ieee.org. An IEEE account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

destruct: *Sanitize* using physical techniques that make recovery of *target data* infeasible using state-of-the-art laboratory techniques and results in the subsequent inability to use the *storage media* for *storage*.

NOTE 1—*Disintegrate*, *incinerate*, and *melt* are *destruct* forms of *media sanitization*.[4]

NOTE 2—If the *storage media* cannot be removed, then the *storage device* can be subjected to the *destruct sanitization* method; a *storage device* can contain multiple instances of *storage media*.

device: Mechanical, electrical, or electronic contrivance with a specific purpose.

deprecated: A *sanitization* method still permitted although its use is discouraged, and it is not guaranteed to be a part of future specification versions.

disintegrate: *Destruct* by separating a *storage device* into its component parts.

host: Computing system that accesses a *storage device* through a *host interface*.

host interface: A component in a *storage device* through which a *host* transfers commands, data, and status.

incinerate: *Destruct* by burning a *storage device* completely.

logical storage: An abstraction of *physical storage* presented at a *host interface*.

logical storage sanitization: *Sanitization* of logical storage.

NOTE 1—*Clear* and *purge* are actions that can be taken to *sanitize logical storage*.

NOTE 2—*Logical storage sanitization* is a subset of *storage sanitization*.

media based cryptographic erase: Method of *cryptographic erase* in which the encryption key is only resident on the *storage device*.

media sanitization: *Sanitization* of *storage media*.

NOTE 1—*Clear*, *purge*, and *destruct* are actions that can be taken to *sanitize storage media*.

NOTE 2—*Media sanitization* is a subset of *storage sanitization*.

melt: *Destruct* by changing *storage media* from a solid to a liquid state, generally by the application of heat.

nonaddressable: A characteristic of data, locations, or physical *storage media* in a *storage device* indicating an inability to be read or written through a *host interface*.

nonvolatile storage: *Storage media* that retains its contents even after power is removed.

physical storage: Physical *storage media*.

pulverize: An obsolete form of *destruct* that grinds a *storage device* to a powder or appropriately small particles.

[4] Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

purge: *Sanitize* using logical techniques or physical techniques that make recovery of *target data* infeasible using state-of-the-art laboratory techniques, but that preserves the *storage media* and the *storage device* in a potentially reusable state.

NOTE—Judicious selection of the *purge* technique increases the likelihood of preserving the *storage device* in a usable state.

sanitization: Process or method to *sanitize*.

sanitize: Render access to *target data* on *storage media* infeasible for a given level of effort.

shred: An obsolete form of *destruct* that cuts or tears a *storage device* or *storage media* into small particles.

storage: *Device*, function, or service supporting data entry and retrieval.

storage device: Any component or aggregation of components made up of one or more *devices* containing *storage media*, designed, and built for the purpose of accessing *nonvolatile storage*.

storage media: Material on which data are or can be recorded or retrieved.

storage sanitization: *Sanitization* of *logical storage* or *storage media*.

store: Record data on *volatile storage* or *nonvolatile storage*.

target data: Information subject to *sanitization*, generally including most or all data on a piece of *storage media*.

user data: *Target data* written and read by users.

volatile storage: *Storage media* that fails to retain its contents after power is removed.

3.2 Acronyms and abbreviations

ATA	AT attachment
BD	Blu-ray disc
CF	CompactFlash
CFast	CompactFlash on serial ATA
CMB	controller memory buffer
DRAM	dynamic random access memory
EAPROM	electrically alterable programmable read-only memory
EEPROM	electrically erasable programmable read-only memory
EEDP	end-to-end data protection
EHM	enable host memory
HDD	hard disk drive
HMB	host memory buffer
ICT	information and communication technology

I/O	input/output
KEK	key encryption key
LBA	logical block address
LTO	linear tape-open
MAM	medium auxiliary memory, a type of nonvolatile data area
MEK	media encryption key
NAND	type of nonvolatile flash memory
NVMe	NVM Express
NVMe-MI	NVMe Management Interface
PATA	Parallel ATA
PI	protection information
PMR	persistent memory region
RAM	random access memory
ROM	read only memory
RPMB	replay protected memory buffer
SATA	Serial ATA
SCSI	small computer storage interface
SED	self-encrypting device or self-encrypting drive
SSC	security subsystem class[5]
SMBus	system management bus
SSD	solid-state drive
SSHD	solid-state hard drive (e.g., hybrid drive)
TCG	Trusted Computing Group
WORM	write once read many

4. Conventions

4.1 Precedence

When a conflict between text, figures, and tables occurs, the precedence shall be (in decreasing order) as follows:

a) text;

b) tables; and

c) figures.

[5] See Annex D.

4.2 Lists

4.2.1 Lists overview

Lists are associated with an introductory paragraph or phrase and are numbered relative to that paragraph or phrase [i.e., all lists begin with an “a)” or an “—” entry].

Each item in a list is preceded by an identification with the style of the identification being determined by whether the list is intended to be an ordered list or an unordered list.

Each item in a list ends with a semicolon, except the last item, which ends in a period. The next-to-the-last entry in the list ends with a semicolon followed by an “and” or an “or” (i.e., “…; and” or “…; or”). The “and” is used if all the items in the list are required. The “or” is used if only one or more items in the list are required.

4.2.2 Unordered lists

An unordered list is one in which the order of the listed items is unimportant (i.e., it does not matter where in the list an item occurs as all items have equal importance).

Each list item shall start with an em dash (“—”). If it is necessary to subdivide a list item further with an additional unordered list (i.e., have a nested unordered list), then the nested unordered list shall be indented and each item in the nested unordered list shall start with an em dash.

The following example shows an unordered list with a nested unordered list:

EXAMPLE—The following items are used for the assembly:

- a box that conntains the following:
  - a bolt;
  - a nut; and
  - a washer;
- a screwdriver; and
- a wrench.

4.2.3 Ordered lists

An ordered list is one in which the order of the listed items is important (i.e., item n is required before item $n+1$).

Each listed item starts with a lowercase Western-Arabic letter followed by a closed parenthesis.

If it is necessary to subdivide a list item further with an additional ordered list (i.e., have a nested ordered list), then the nested unordered list shall be indented and each item in the nested unordered list shall start with a decimal number followed by a closed parenthesis.

If it is necessary to subdivide a list item further with an additional unordered list (i.e., have a nested unordered list), then the nested unordered list shall be formatted as specified in 4.2.2.

The following example is of an ordered list with a nested unordered list and a nested ordered list.

EXAMPLE—The instructions for the assembly are as follows:

a) remove the contents from the box;
   - unwrap each item; and
   - check the inventory;

b) assemble the item;
   - use a screwdriver to tighten the screws; and
   - use a wrench to tighten the bolts;

   and

c) take a break.

4.3 Numbering

A binary number is represented in this standard by any sequence of digits consisting of only the Western-Arabic numerals 0 and 1 immediately followed by a lowercase b (e.g., 0101b). Underscores or spaces may be included between characters in binary number representations to increase readability or delineate field boundaries (e.g., 00101 1010b or 0_0101_1010b).

A hexadecimal number is represented in this standard by any sequence of digits consisting of only the Western-Arabic numerals 0 through 9 and/or the uppercase English letters A through F immediately followed by a lowercase h (e.g., FA23h). Underscores or spaces may be included between characters in hexadecimal number representations to increase readability or delineate field boundaries (e.g., B FD8C FA23h or B_FD8C_FA23h).

A decimal number is represented in this standard by any sequence of digits consisting of only the Arabic numerals 0 through 9 not immediately followed by a lowercase b or lowercase h (e.g., 25). This standard uses the following conventions for representing decimal numbers:

- the decimal separator (i.e., separating the integer and fractional portions of the number) is a period;
- the thousands separator (i.e., separating groups of three digits in a portion of the number) is a space; and
- the thousands separator is used in both the integer portion and the fraction portion of a number.

Table 1 shows numbering conventions for this standard with comparisons to other national conventions.

Table 1—Numbering conventions

French	English	This standard
0,6	0.6	0.6
3,141 582 65	3.14158265	3.141 582 65
1 000	1,000	1 000
1 323 462,95	1,323,462.95	1 323 462.95

A decimal number represented in this standard with an overline under one or more digits following the decimal point is a number where the overlined digits are infinitely repeating (e.g., 666. $\overline{6}$ means 666.666 666… or 666 2/3, and 12. $\overline{142857}$ means 12.142 857 142 857… or 12 1/7).

4.4 Bit conventions

n:m, where n is greater than m, denotes a set of bits [e.g., Feature (7:0)]. n:m where n is greater than m, denotes a bit range.

4.5 Number range convention

p..q, where p is less than q, represents a range of numbers (e.g., bytes 100..103 represents 100, 101, 102, and 103).

4.6 Small caps

A word is written in SMALL CAPITALS if it occurs in SMALL CAPITALS in a referenced standard or specification.

5. Storage sanitization

5.1 General

ICT systems capture, process, and *store* data using a wide variety of storage. These data are not only located on the intended *storage media* but also on *storage devices* used to *store*, process, or transmit this information. These *storage media* can require special disposition to mitigate the risk of unauthorized disclosure of data (e.g., to help ensure the confidentiality of that data). Efficient and effective management of data created, processed, and stored by an ICT system throughout its life, from its inception through disposition, is a primary concern of an ICT system owner and the custodian of the data.

When *storage devices* are transferred, become obsolete, are no longer usable, or required by an ICT system, it is important that residual magnetic, optical, electrical, or other representation of data are not recoverable. For sensitive or regulated data, controlled elimination of data recorded on *storage media* is a necessity. *Storage sanitization*, henceforth *sanitization*, refers to the general process of denying access to data from *storage media*, such that reasonable assurance exists that the data cannot be retrieved or reconstructed. The focus on access is important because sometimes the data on the *storage media* cannot be eliminated, so other steps (e.g., destruction of the *storage media*) can become necessary.

If the *sanitization* of data is intended to remove all instances of specific data, then all media on which that data has been stored (e.g., as a result of caching, replication, mirroring or other redundancy, backup or point in time copies, swapping, and paging) also requires *sanitization*.

An example of this is NVMe HMB data stored in persistent memory (e.g., NVDIMM) or swapped to backing storage.

The concept of *sanitization* of data recorded on *storage media* is easy to understand; however, putting the concept into practice can be challenging. An additional complication is the inconsistent use of vague terminology to describe this elimination of data. Vague terms with poorly defined meanings include the following:

— deletion, which can refer to the file system operation that removes a few file system pointers (e.g., no data are removed);

— secure data deletion, which potentially only removes currently accessible copies of data as opposed to all copies of data on the *storage media*; and

— data shredding, which can refer to a physical *shredder* or scrambling of returned data by the deletion of the encryption key.

Annex C describes some emerging technologies out of scope for this standard.

5.2 Elements of sanitization

Sanitization is often an important part of an organization's data governance program, which generally includes policies and processes that focus on handling sensitive data (e.g., personal data, personally identifiable information, electronic healthcare records, trade secrets, intellectual property, customer records, financial records, or mission-critical data). Such a governance program covers the full data lifecycle (creation/generation/collection, processing, transferring, storing, archiving, and destruction of data) and the ICT infrastructure associated with these data. An important aspect of this governance addresses the need to eliminate data due to changes in business needs or compliance obligations, which can include producing appropriate documentation to serve as evidence of the actions taken.

Eliminating data, or rendering data permanently inaccessible, can be handled by one or more of the following:

— data *sanitization*: Focused on all instances of stored data, wherever the data resides. Such elimination can be quite challenging because all copies need to be identified, and adequate data maps potentially do not exist. These data copies can exist within applications, cloud services, virtual environments, primary compute and storage resources, secondary and off-line storage, archives, and data protection systems (e.g., backups and replications). For each of these data locations, specific technology-oriented actions (to achieve *storage sanitization*) are then necessary to eliminate the data;

— *storage sanitization*: Focused on data stored on ICT infrastructure that uses *nonvolatile storage* (e.g., fixed-block *storage* arrays, network attached storage systems, object storage, cloud storage, and backup systems) that can take the form of *logical storage* or physical storage that contains *storage media*; and

— *media sanitization*: Focused on data stored on *storage devices* or *storage media*.

Sanitization can involve some or all of the following:

— identifying the type of storage involved: logical *storage* or *media* aligned (*media sanitization*);

— selecting the *sanitization* method (i.e., *clear*, *purge*, or *destruct*) appropriate for the type of *storage device* and the data sensitivity;

— executing one or more of the selected *storage sanitization* techniques;

— verifying the results of the *storage sanitization* to determine the level of residual risk (see Clause 7); and

— producing evidence of the storage sanitization that meets compliance obligations (proof of *sanitization*).

This standard primarily provides guidance and requirements for *media sanitization* and secondarily provides limited discussions of requirements for level of effort for data recovery. This standard does not address requirements for proof of *sanitization* or requirements for verification of *sanitized* storage. However, issues associated with each of those are addressed when appropriate.

5.3 Conformance

For purposes of conformance, the choice of *sanitization* method (see Clause 6) is important because the level of protection can vary significantly. The three *sanitization* methods in this standard result in varying levels of effort being required to recover *target data* after *sanitization* has been performed, with the *clear sanitization* method requiring the least effort and the *destruct sanitization* method requiring the most effort.

Conformance with this standard shall be based on the specific *sanitization* methods (*clear*, *purge*, or *destruct*) applicable for a particular *storage media* type (see Clause 8). When the *sanitization* requirement does not specify a *sanitization* method, conformance shall be achieved through the use of any applicable *sanitization* method for the *storage media* type. When the *sanitization* requirement specifies a *sanitization* method, conformance shall be achieved by using the specified sanitization method applicable for the *storage media* type or by using a sanitization method requiring a greater effort for attempted data recovery.

For example, an organization can designate *clear* as an adequate *sanitization* method without reference to the *storage media* type. However, *clear* and *purge* do not apply to paper hardcopy; therefore, *destruct* is the only conforming *sanitization* method for paper hardcopy.

5.4 Accessibility

An organization wishing to sanitize *media* first determines whether it can apply the procedures defined in this standard. The following factors affect the organization's ability to sanitize some types of *storage media*:

— the *storage media* is not identifiable. For example, while cartridges usually are labeled with the technology and generation, some are not labeled;

— the organization lacks the expertise to sanitize the *storage media* (while leaving it usable) or to verify (see Clause 7) that *sanitization* was successful;

— the equipment is not working or is anticipated to not be working soon; and

— the equipment or software needed to perform the operations is not available. Examples include a *storage device* to access removable *storage media*, an interface for the *storage device*, a degausser with sufficient strength to erase newer magnetic *storage media*, and so on.

If the organization cannot sanitize the *storage media* and cannot locate another organization that can do so, then the *storage media* shall be destroyed using the *destruct sanitization* method appropriate to the *storage media* type. See Clause 6 for definitions of *sanitization* procedures.

5.5 Sustainability and media sanitization

Companies have requirements to keep customer data safe or internal data from leaking, as well as external commitments to the environment and sustainability. Selecting the proper form of media sanitization with data sensitivity also needs to be weighed with the consequences of *sanitization* methods that render the device unusable.

Circular business models are essential for the ICT industry as a large percentage of energy and carbon emissions come from manufacturing products. Circularity business models design waste out, keep materials in use as long as possible, and restore environmental systems in the process.

Companies should prioritize the *purge sanitization* method over the *destruct sanitization* method for most data and work with their vendors to qualify and verify the *purge sanitization* method.

6. Sanitization methods and techniques

6.1 General

The *clear*, *purge*, and *destruct sanitization* methods can be employed to *sanitize physical storage*. *Sanitization* of *logical storage* is not addressed by this standard. These *sanitization* methods use techniques specific to the type of *storage media* being *sanitized* (see Clause 8). This clause describes each *sanitization* method and provides additional options where appropriate. Table 2 describes the general principles encompassed by each *sanitization* method. Clause 8 contains additional information specific to each *media* type.

Table 2—General sanitization method comparison

Element	Description	Result of Sanitation Method		
		Clear	**Purge**	**Destruct**
Usability	Is the *storage device* usable after *sanitization*?	Yes.	It depends on the specific *purge* technique used.	No.
User data	Data that can be read from the *storage media* using the *host interface*.	Previous contents are no longer retrievable using simple noninvasive data recovery techniques.	Previous contents are no longer retrievable using state-of-the-art laboratory techniques.	Previous contents are no longer retrievable using state-of-the-art laboratory techniques.
Target data that was *user data*	*Storage media* that previously held *user data* but no longer accessible to the *host* because of reallocation, uncorrectable *storage media* errors, wear leveling, etc.	May or may not be changed.	Previous contents are no longer retrievable.	Previous contents are no longer retrievable.
Target data that could become *user data*	*Storage media* not accessible to the *host* and that has not yet been used for *user data*, but may be written with *user data* if any *user data* is reallocated.	May or may not be changed.	May or may not be changed.	Previous contents are no longer retrievable.
Other	System areas, firmware, logs, etc.	Are not changed.	May or may not be changed.	Previous contents are no longer retrievable.

The choice of which *sanitization* method to use depends on the owner of the data, the *storage media*, and the organizational security policy to be enforced. Some security policies may require *clear*, whereas others may require *purge* or *destruct*. For example, some organizational policies may favor *purge* over *destruct* for environmental concerns (see 5.5).

Sanitization methods can fail for various reasons. If a *clear sanitization* method fails, then retry an alternative *clear sanitization* method until successful or use the *purge sanitization* method. If a *purge sanitization* method fails, then retry an alternative *purge sanitization* method until successful or use the *destruct sanitization* method.

The *sanitization* method to use depends on the organizationally determined level of security categorization, as well as on other factors. An example of a security categorization is as follows:

— Low (e.g., *clear*): Information if disclosed to an unauthorized party would cause mild impacts to the organization;

— Medium (e.g., *purge*): Information if disclosed to an unauthorized party would cause moderate impacts to the organization; and

— High (e.g., *destruct*): Information if disclosed to an unauthorized party would cause severe impacts to the organization.

Figure 1 shows the *sanitization* process, which begins with the selected *sanitization* method.

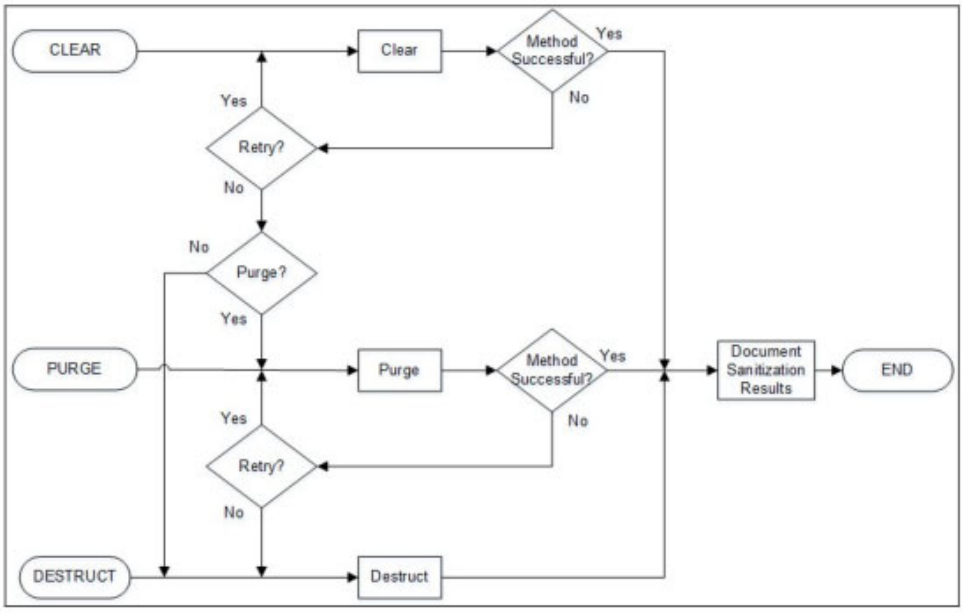

Figure 1—Sanitization process

6.2 Clear

The *clear sanitization* method uses logical techniques on all *addressable storage* locations for protection against simple, noninvasive data recovery techniques using the same *host interface* available to the user. Data can be replaced through the *host interface* using the appropriate *storage device* command(s) to modify the data on all *addressable* locations.

Clear is not appropriate for sensitive data because *clear* is not required to remove data from *nonaddressable* locations. For moderate confidentiality data, the *storage media* owner can choose to accept the risk of applying the *clear sanitization* method to the *storage media*, acknowledging that some data are retrievable by someone with the time, knowledge, and skills to do so.

For example, when data at a logical address are replaced on a solid-state drive (SSD), the new data are written to a different *storage media* location. Reading that logical address returns the new data. However, old data potentially remains on the *storage media* at its old physical address. If the *storage device* were disassembled in a laboratory and the *storage media* accessed at its physical addresses, the original data could be recovered. For this reason, a *storage device* with sensitive *user data* should be *sanitized* with the *purge sanitization* method before being made accessible to a different user who is not authorized to access the previously *stored* sensitive *user data*.

Techniques for performing *clear* include overwrite (see 6.5.1) and block erase (see 6.5.2).

6.3 Purge

The *purge sanitization* method uses logical techniques or physical techniques that make recovery of *target data* infeasible using state-of-the-art laboratory techniques applied to an intact or a disassembled *storage device* but that preserves the *storage media* and the *storage device* in a potentially reusable state.

Judicious selection of the *purge* technique increases the likelihood of preserving the storage device in a usable state. State-of-the-art laboratory techniques include *storage device* component access (e.g., placing an HDD platter in a spin stand) and mechanical creation of *storage device* component access (e.g., removing the top layers of an integrated circuit package, aka “decapping,” to expose the integrated circuits).

Techniques for performing *purge* include *sanitization* using overwrite (see 6.5.1), *sanitization* using block erase (see 6.5.2), and *media-based cryptographic erase* (see 6.5.3), when applied to all *addressable* and *nonaddressable* physical *storage media*.

Degaussing (see 6.5.4) may be an acceptable *purge sanitization* method given appropriate vendor documentation and vendor-approved tools.

6.4 Destruct

The *destruct sanitization* method makes recovery of *target data* infeasible using state-of-the-art laboratory techniques and results in the subsequent inability to use the *storage media*.

The techniques for *storage media* destruction are as follows:

— **disintegrate:** *Sanitization* method designed to completely destroy the *storage media* by breaking or decomposing (e.g., dissolving with acid) it into its constituent elements, parts, or small particles;

— **incinerate:** *Sanitization* method designed to completely destroy the *storage media* by burning until it is reduced to ashes; and

— **melt:** *Sanitization* method designed to completely destroy the *storage media* by liquefying it, generally through the application of heat.

Although *pulverize* and *shred* were once adequate forms of *destruct*, improvements in reconstruction technology and increases in the density of information on the *storage media* have rendered these techniques

ineffective for *storage media* other than for low-density *storage media* (e.g., hardcopy and floppy disks). Use of *pulverize* and *shred* should be considered carefully based on the *storage media* under consideration. It is acceptable to incorporate *pulverize* and *shred* methods as an interim destruction step prior to shipping *storage media* for further processing such as *melt*, *incineration*, or *disintegration*.

Depending on the material, these techniques can generate hazardous materials (e.g., toxic dust or combustion products). Environmental protection regulations should be considered before performing the techniques and disposing of the by-products.

6.5 Clear and purge techniques

6.5.1 Sanitization using overwrite

Overwriting applies to most electronic *storage media* types. It is not applicable to *storage media* that do not permit data to be erased or changed after being written (e.g., WORM tape cartridges). Because every *addressable* location (e.g., logical blocks or sectors) in the *storage device* is altered, overwriting a large-capacity *storage device* can take a significant amount of time.

Writing a pattern to all *addressable* locations (e.g., overwrites data accessible to the *host*) is overwriting as a *clear sanitization* method. Using a *device* command that writes to locations *addressable*, and to locations not *addressable* (e.g., reallocation pools, overprovisioning, and caches), is overwriting as a *purge sanitization method*.

Overwriting is problematic for *storage media* damaged or not rewriteable.

6.5.2 Sanitization using block erase

Block erase generally applies to semiconductor *storage media* (e.g., NAND flash allows a large region of a *storage*, known as an “erase block,” to be erased in a single operation). Although this operation can be faster than overwriting, block erasing a large-capacity *storage device* containing many erase blocks can nevertheless take a significant amount of time, although much less than overwriting.

Block erase is performed by issuing commands through the *host interface*. For example, many *storage device* types implement format or *sanitize* commands, which can specify that all accessible and inaccessible erase blocks capable of containing *user data* in the *storage device* are erased. Commands to block erase individual blocks are not generally implemented in standard *storage device* command sets.

6.5.3 Media-based cryptographic erase

Storage media based on *cryptographic erase* applies to electronic *storage media* types that contain data encrypted using encryption keys that reside in the *storage device* and can be changed in a single operation. Changing the keys leaves only the ciphertext remaining on the *storage media*, effectively *sanitizing* the data.

Storage media based on *cryptographic erase* cannot be performed at the *storage device* level in encrypted electronic *storage media* types (e.g., encrypted LTO-4 tapes) in which the encryption keys are not stored on the *storage media*.

Storage media based on *cryptographic erase* is performed by issuing commands through the *host interface* to change the encryption keys. For example, many *storage devices* types implement format or sanitize commands that can specify that a *cryptographic erase* is performed.

Keys shall be randomly generated from the entire keyspace.

To use *cryptographic erase* as a *purge sanitization method*, the following conditions shall be met at a minimum:

— encryption of all data intended for *cryptographic erase* prior to recording on the *storage media*;

— the strength of the cryptographic algorithm (including mode of operation) used to encrypt the *target data* is at least 128 bits;

— the level of entropy of the encryption key used to encrypt the *target data* is at least 128 bits; and

— all copies of the encryption keys used to encrypt the *target data* are *sanitized*; if the *target data*'s encryption keys are, themselves, encrypted with one or more wrapping keys, it is acceptable to perform *cryptographic erase* by *sanitizing* a corresponding wrapping key.

NOTE—Although it might be tempting to combine *cryptographic erase* with another *sanitization* method (e.g., *clear*), such an approach does not improve security, but it can significantly slow the *sanitization* operation and potentially impede the ability to verify the *cryptographic erase*. Justifications for such an approach often include efforts to reduce the attack surface by preventing access to the ciphertext, but this simply highlights that *cryptographic erase* is not appropriate for the sensitivity level of the data.

For additional information on *cryptographic erase*, see Annex B.

6.5.4 Degaussing

Degaussing applies to magnetic *storage media*. It does not apply to *storage devices* that contain nonmagnetic *storage media* (e.g., paper, SSD, or the nonmagnetic components in an SSHD).

Degaussing exposes the magnetic *storage media* to a strong magnetic field to disrupt the recorded magnetic domains. A degausser *device* generates a magnetic field used to sanitize magnetic *storage media*. Degaussers are rated based on the strength of their generated magnetic field, which may limit the types of magnetic *storage media* that they can *purge*. Degaussers operate using either a strong permanent magnet or an electromagnetic coil. *Degaussing* can be an effective technique for purging damaged or inoperative *storage media*, for purging *storage media* with exceptionally large *storage* capacities, or for quickly purging diskettes.

Generally, newer types of magnetic *storage media* can require higher field strengths to achieve *degaussing* than is required for older types of *storage media*. Thus, some degaussers cannot *purge* some *storage media* (e.g., a degausser only sufficient to *purge* an LTO-1 tape cartridge is not capable of *purging* an LTO-7 tape cartridge). It is essential to understand whether a particular degausser can *purge* a particular type of *storage media*. If a degausser is marginally capable of purging a particular *storage media* type, then some areas of the *media* could be completely erased while data retrieval remains possible from other areas.

If the applicability or effectiveness of *degaussing* as a *purge sanitization method* cannot be determined, then use an alternative *purge sanitization method* or the *destruct sanitization* method.

For certain *storage media* types (e.g., tapes with servo tracks and HDD), *degaussing* with sufficient field strength renders the *storage media* unusable when sufficient is defined as a field strength that exceeds the coercivity by enough to completely erase any stored data. If the field strength is insufficient, then it is possible that the *storage device* is unusable but *target data* remain on the *storage media*.

Degaussing is only an acceptable *destruct sanitization* method when following appropriate vendor documentation and using vendor-approved tools.

7. Verification of sanitization outcomes

7.1 General

Verification of the *sanitization* outcomes can be an important element of a data *sanitization* program when a determination as to the adequacy or effectiveness of the *storage sanitization* is required. This verification differs depending on the *sanitization* method. For *clear* or *purge*, the *storage device* interface is used to check the results of the *sanitization* operation. For *destruct*, physical inspection is used to check the *sanitization* outcomes. Verification is important because errors or anomalies can necessitate additional actions to complete the *sanitization* or a decision on the part of the organization to accept any residual risk.

Verification can consist of either verifying the successful functioning of a *sanitization* operation (e.g., the size of particles produced by a *shredder*) or verification that a command was performed (but not the actual functioning of the command). This standard does not require verification of the actual functioning of the command by any particular implementation. Some aspects of verification require destruction, disassembly, or state-of-the-art laboratory techniques beyond the capability of most organizations.

For the *clear sanitization* method, verification (see 7.2) using representative sampling (see 7.3) should be performed, assuming that any form of verification is deemed necessary.

For the *purge sanitization* method, a full verification (see 7.2) of the *addressable storage media* should be performed. If *cryptographic erase* was used to perform the *sanitization*, it is potentially not possible to perform verification (see 7.4).

It is important to note that *devices* protected with access control mechanisms have additional verification considerations. Whether such *devices* were *sanitized* by overwrite, block erasing, or *cryptographic erase*, such *devices* need to be accessible before and after *sanitization* to enable a verification process.

For the *destruct sanitization* method, physical inspection (see 7.5) is the only option because the *storage* has (by definition) been made unusable.

The findings from the verification can result in further *sanitization* activities when the *sanitization* outcomes are not adequate. In addition, there can be a need to record the findings from the verification, but this is out of scope for this standard.

Sanitization can leave the *device* in a state where it is not possible to read data from the *storage media* without error.

Some *purge sanitization* methods (i.e., block erasure and cryptographic erasure) can also affect data coherency information (e.g., cyclic redundancy check, integrity, or protection information), such that the *storage media*, if read, cannot return valid *user data* until written after the sanitization method has completed. Such implementations generally do not allow uncorrectable data through the data path. Such implementations can return data associated with deallocation (e.g., zeros) until the *storage media* is written again with valid coherency data. Verification of such implementations can only check that the original *user data* are no longer returned across the *host interface*.

Specifications that require *storage media* verification after *sanitization* methods such as cryptographic erasure and block erase pose problems for *storage devices* that cannot read *storage media* without error after such a technique is performed. An internal full *storage media* overwrite could be necessary to enable

reading *storage media*, or a *host interface* extension could be provided to read *storage media* without checking for errors.

7.2 Full verification

A full verification that compares read data with a single expected *sanitized* value is achieved by a full reading of all areas to be verified and comparing the read data with the expected *sanitized* value. This manner of verification generally only applies where the *storage device* is in an operational state following *sanitization* so that data can be read and written through the *storage device host interface*. Note that full verification cannot verify areas inaccessible through the *storage device host interface*.

If using *cryptographic erase*, then full verification is not effective because the resulting data are unpredictable if not deallocated. For guidance on verification for media-based cryptographic erase, see 7.4.

Full verification is also not effective for some block erase implementations for the same reason (resulting data are unpredictable if not deallocated). The guidance in 7.4 also applies to block erase implementations that exhibit this behavior.

7.3 Representative sampling

If an organization chooses representative sampling, then one of the following options should be used to perform electronic *storage media sanitization* verification:

— select random locations on the *storage media* that represent at least 5% of the *addressable* space; or

— select locations across the *addressable* space. For instance, conceptually break the *storage media* up into equal-sized subsections. Select a large enough number of subsections so that the *storage media* is well covered. The number of practical subsections depends on the *storage device* and addressing scheme. The suggested minimum number of subsections for *storage devices* leveraging LBA addressing is 10 000. Select at least two nonoverlapping pseudo-random locations from within each subsection. For example, if 10 000 conceptual subsections are chosen, at least two random locations in the first 1 / (10 000) of the *storage media addressable* space would be read and verified, at least two random locations in the second ten-thousandth of the *storage media addressable* space would be read and verified, and so on. In addition to the locations already identified, include the first and last *addressable* locations on the *storage device*.

7.4 Verification for media-based cryptographic erase

Media-based cryptographic erase has different verification considerations than other procedures because the contents following *cryptographic erase* are not known and therefore cannot be compared with an expected value. When *cryptographic erase* is leveraged, an attempt should be made to apply simple verification checks such as reading a *storage media* location with known contents to verify that the expected data are not returned. If it is not possible, for whatever reason (e.g., person executing *cryptographic erase* does not have read access), then verification can be skipped if allowed by organizational policy.

The guidance in this clause also applies to block erase implementations for which the contents following block erase are not known.

7.5 Verification by physical inspection

Physical inspection is the only option when *destruct* is the *sanitization* method because the *storage device* (by definition) has been made unusable. If, after reviewing the verification findings associated with the *destruct* outcomes a determination is made that the sanitization outcomes are not adequate, then the *destruct*-based *sanitization* should be repeated with consideration given to using an alternative form of *destruct*.

8. Media type-specific sanitization

8.1 General

For some *storage devices*, data can be *cleared* using a single overwrite pass with a fixed pattern (e.g., zeroes) to prevent recovery of data even if state-of-the-art laboratory techniques are applied.

One major drawback of relying solely on the *host interface* for performing the overwrite procedure is that areas not currently *addressable* (e.g., defect areas and currently unallocated space) are not overwritten. Dedicated sanitize commands support changing these areas more effectively. The use of such commands results in a trade-off because although they can more thoroughly change all areas of the *storage media*, using these commands also requires trust and assurance from the vendor that the commands have been implemented as expected.

Users who have become accustomed to relying on overwrite techniques, and who have continued to apply these techniques as *storage media* types evolved, can be exposing their data to increased risk of unintentional disclosure. Although the *host interface* (e.g., ATA, SCSI, or NVMe) can be similar across *storage devices* with varying underlying *storage media* types, it is critical that the *sanitization* techniques are carefully matched to the *storage media* type.

Some destructive techniques for some *media* types can become more difficult or impossible to apply in the future. Traditional techniques become more complicated as *storage media* evolves. For example, emerging variations of magnetic recording technologies incorporate *storage media* with higher coercivity (magnetic force), and some degaussers do not have sufficient force to effectively *degauss* such *storage media*.

Applying destructive techniques to nonmagnetic *storage media* (e.g., semiconductor and optical) is also becoming more challenging as the necessary particle size for commonly applied grinding techniques goes down proportionally to any increases in *storage* density. Semiconductors already present challenges with occasional damage to grinders due to the hardness of the component materials, and this problem becomes worse as grinders attempt to grind the components into even smaller pieces.

Table 3 describes the *storage media* types discussed in this standard.

Table 3—Storage media types in this standard

Storage media type	Reference
Hard copy	8.2
Optical	8.3
HDD, SSHD, and SSD (ATA, SCSI, and NVMe)	8.4
Other magnetic	8.5
USB removeable	8.6
Memory cards	8.7
Embedded flash on boards and *storage devices*	8.8
RAM and ROM-based *storage devices*	8.9
Developing storage technologies	Annex C

8.2 Hard copy

Hard-copy *storage media* are physical representations of information, most often associated with paper printouts. The supplies associated with producing paper printouts are often the most uncontrolled. Hard-copy materials containing sensitive data that leave an organization without effective *sanitization* expose a significant vulnerability to "dumpster divers" and overcurious employees, risking accidental disclosures. Guidance for this type of *storage media* can be found in 8.2.3.

Paper, microforms (microfilm, microfiche, or other reduced-image photo negatives), printer and facsimile ribbons, drums, and platens are examples. (See Table 4.)

Table 4—Paper and microforms

Sanitization method	Reference
Clear	8.2.1
Purge	8.2.2
Destruct	8.2.3

8.2.1 Clear

Only *destruct* (see 8.2.3) is acceptable.

8.2.2 Purge

Only *destruct* (see 8.2.3) is acceptable.

8.2.3 Destruct

Destruct paper using crosscut *shredders* that produce particles 1 × 5 mm in size or smaller or that *pulverize/disintegrate* paper materials using a *disintegrator device* equipped with a 1.5 mm security screen.

Destruct microforms (microfilm, microfiche, or other reduced-image photo negatives) by burning. When material is burned, the residue is reduced to ash.

See 6.5.4 for additional considerations.

8.3 Optical media

8.3.1 CD, DVD, Blu-ray

Optical *storage media* like CD, DVD, and Blu-ray allow read-only access or read-write access (Table 5).

Table 5—CD, DVD, Blu-ray sanitization

Sanitization method	Reference
Clear	8.3.1.1
Purge	8.3.1.2
Destruct	8.3.1.3

8.3.1.1 Clear

Only *destruct* (see 8.3.1.3) is acceptable.

8.3.1.2 Purge

Only *destruct* (see 8.3.1.3) is acceptable.

8.3.1.3 Destruct

The following *destruct* techniques are listed in order from the most preferable to the least preferable:

a) remove the information-bearing layers of CD *storage media* using a commercial optical disk grinding *device*. Note that this applies only to CD and not to DVD or BD *storage media*; or

b) *incinerate* optical disk *media* (reduce to ash) using a licensed facility; and

c) use optical disk *storage media shredders* or *disintegrator devices* to reduce to particles that have nominal edge dimensions of 0.5 mm and surface area 0.25 mm^2 or smaller.

8.4 HDD, SSHD, and SSD (ATA, SCSI, and NVMe) storage

8.4.1 Overview

A single *storage device* using the ATA, SCSI, or NVMe command sets can contain the following types of *storage media*:

— magnetic;

— volatile memories; and/or

— nonvolatile memories.

Subclause 8.4 applies to all HDD, SSHD, and SSD *storage devices* that use the command set families listed in Table 6.

Table 6—Transports for ATA, SCSI, and NVMe command set families

Command set family	Transports
ATA	Parallel ATA (PATA), Serial ATA (SATA), eSATA, CompactFlash, CFast
SCSI	Parallel SCSI, SAS, USB, UAS, IEEE 1394 (FireWire), ATAPI, Fibre Channel,[a] iSCSI, UFS
NVMe	PCIe, TCP, RDMA, Fibre Channel

[a] The use of Fibre Channel with other *host interfaces* is out of scope for this standard.

These *storage devices* can be installed internally to a host system, externally to a *host* system (e.g., via cabling), or as part of an enclosure.

Some *storage devices* can have a management interface capable of initiating *sanitization*.

The drive could be configured in a vendor-specific manner that precludes *sanitization* when removed from the enclosure. In this case, follow appropriate vendor documentation and use vendor-approved tools.

Specific recording techniques affect *storage media* types differently. The useable lifetime of NAND *storage media* is reduced by overwriting it excessively, and it is less affected by the block erase *sanitization* technique (see 6.5.2). Magnetic *storage media* are designed for extensive overwriting, and they cannot be block erased. Solid-state memories are not affected by *degaussing*.

If a *storage device* has multiple types of *storage media* (e.g., magnetic and NAND), then the method of *sanitization* depends on the *storage media* type (see 8.4.3 and 8.4.4).

A *storage device* can include security technology from TCG that provides support for *sanitization* methods. This is independent of all transport types (e.g., ATA, SCSI, and NVMe).

Refer to the standards and specifications listed in Table 7 for details on specific commands referenced in 8.3. The full document titles and references are listed in Annex D.

Table 7—Relevant standards and specifications for ATA, SCSI, NVMe, and TCG[6]

ATA	SCSI	NVMe	TCG
ACS-2 [B8]	SPC-5 [B3]	NVM Express Base Specification Revision 2.0b [B10]	Enterprise SSC [B15]
ACS-5 [B7]	SBC-4 [B4]	NVM Express NVM Command Set Specification Revision 1.0b [B12]	Opal SSC [B17], [B18]
ZAC-2 [B5]	ZBC-2 [B6]	NVM Express Zoned Namespace Command Set Specification Revision 1.1b [B13]	Opalite SSC [B19]
		NVM Express Key Value Command Set Specification 1.0b [B11]	Ruby SSC [B22]
		NVM Express Management Interface Specification Revision 1.2a [B14]	Pyrite SSC [B20], [B21]
			Storage Interface Interactions Specification [B23]

[6] The numbers in brackets correspond to those of the bibliography in Annex D.

Some *storage devices* may have hidden *storage media* areas not *addressable*. The *storage device* vendor may use organizationally approved proprietary commands to interact with the security subsystem. Please refer to the manufacturer to identify whether such areas exist on the *storage media* and whether any tools are available to remove or sanitize them, if present.

Table 8 shows how different types of data within the *storage device* are or are not affected by each *sanitization* method. Use this information, in addition to Clause 6, to select the *sanitization* method with the desired results.

This standard does not specify verification techniques using the *host interface* for the concerns in Table 8, except in the following cases:

— *User data*; and

— CMB.

Table 8—Choosing sanitization method for ATA, SCSI, and NVMe

Concern	Applicable interface	Description	Result of sanitization method		
			Clear (see 8.4.2)	**Purge (see 8.4.3)**	**Destruct (see 8.4.4)**
Usability	All	Is the *storage device* useable after *sanitization*?	Yes.	Yes.	No.
User data	All	Data that can be read from the *storage device* to the *host* including PI and EEDP (if any).	Previous contents are no longer retrievable. See NOTE 1. See NOTE 2.	Previous contents are no longer retrievable.	Previous contents are no longer retrievable.
Previous *target data*	All	Data that were previously "current *user data*" but are no longer accessible to the *host* because of reallocation, uncorrectable *storage media* errors, wear leveling, etc.	Not discernable.	Previous contents are no longer retrievable.	Previous contents are no longer retrievable.
Potential *target data*	All	Overprovisioned *storage* not accessible to the *host* that has not yet been used for *user data*, but may become used for current *user data* if any current *user data* is reallocated.	Not discernable.	Previous contents are no longer retrievable.	Previous contents are no longer retrievable.
Target data in volatile caches	All	*User data* in volatile caches, cleared by power cycle.	Not discernable.	Previous contents are no longer retrievable.	Previous contents are no longer retrievable.
Target data in nonvolatile caches	All	*User data* in nonvolatile caches.	Not discernable.	Previous contents are no longer retrievable.	Previous contents are no longer retrievable.

Concern	Applicable interface	Description	Result of sanitization method		
			Clear (see 8.4.2)	**Purge (see 8.4.3)**	**Destruct (see 8.4.4)**
Saved values of nonvolatile data	All	Set features, mode pages, configuration data, etc.	Depends on implementation.	Depends on implementation.	Previous contents are no longer retrievable.
Write-protected *user data*	All	Current *user data* that can be read but not written.	Not changed.	Previous contents are no longer retrievable.	Previous contents are no longer retrievable.
Write-protected platform data	All	Platform data required for functionality (e.g., mobile *device* communication protocol initialization and configuration data).	Not changed.	Not changed.	Previous contents are no longer retrievable.
Firmware	All	Firmware that the *storage device* uses.	Not changed.	Not changed.	Previous contents are no longer retrievable.
Feature settings	All	Feature settings (e.g., write cache enabled, features supported/not supported).	Not changed.	Not changed.	Previous contents are no longer retrievable.
Logs	All	Logs that do not contain user data [e.g., SMART data, and telemetry (firmware crash dumps)].	Depends on implementation.	Depends on implementation.	Previous contents are no longer retrievable.
Capacity	All	Available storage capacity.	Not changed.	Capacity can be reduced if *storage media* was permanently removed from service during the *sanitize* operation.	N/A
TCG credentials	All	Credentials.	Not changed.	Previous contents are no longer retrievable.	Previous contents are no longer retrievable.
Boot partition	NVMe	*User data*, network data, boot code.	Not changed. See 8.4.2.5.1 for specific requirements.	Not changed. See 8.4.3.4.1 for specific requirements.	Previous contents are no longer retrievable.
CMB	NVMe	Volatile lookup tables, queues, PRP/SGL lists, *user data*.	Previous contents are no longer retrievable.	Previous contents are no longer retrievable.	Previous contents are no longer retrievable.

Concern	Applicable interface	Description	Result of sanitization method		
			Clear (see 8.4.2)	**Purge (see 8.4.3)**	**Destruct (see 8.4.4)**
HMB	NVMe	Region of host memory used by the controller that may contain user data.	Previous contents are no longer retrievable.	Previous contents are no longer retrievable.	Previous contents are no longer retrievable.
PMR	NVMe	*Nonvolatile user* data, system metadata.	Not changed.	Previous contents are no longer retrievable.	Previous contents are no longer retrievable.
NVMe-oF In-band authentication credentials	NVMe	Credentials.	Not changed.	Not changed.	Previous contents are no longer retrievable.
NOTE 1—Results vary by which TCG method is used (if any). NOTE 2—For NVMe: *Clear* affects only attached namespaces. To *sanitize user data* in detached namespaces, either attach them prior to performing the *clear sanitization* method or use the *purge* or *destruct sanitization* methods.					

8.4.2 Clear

8.4.2.1 General

The *clear sanitization* method overwrites *storage media* by using organizationally approved overwriting technologies, techniques, or tools.

The *clear sanitization* method consists of performing the operations specified in the following subclauses:

— 8.4.2.2 for ATA *storage devices*;

— 8.4.2.4 for SCSI *storage devices*; and

— 8.4.2.5 for NVMe subsystems.

In addition to the operations above, the *clear sanitization* method can also include vendor-specific operations.

If it is required to remove the data in the TCG MBR table or the TCG Datastore tables, then the *purge* (see 8.4.3) or the *destruct* (see 8.4.4) *sanitization* method shall be used instead of the *clear sanitization* method.

If, for any element of a *storage device*, any operation in the *clear sanitization* method fails, or if verification fails, then the *sanitization* method has failed. The *clear sanitization* method should be retried until successful; otherwise, the *purge* or *destruct sanitization* method should be used.

8.4.2.2 Clear for ATA

To perform the *clear sanitization* method for ATA, then the *host* shall do the following:

a) reset configuration options that limit the access to portions of the *storage media*, such as follows:

— Host Protected Area (HPA) (see ISO/IEC 17760-102:2016 [B8]);

— Device Configuration Overlay (DCO) (see ISO/IEC 17760-102:2016 [B8]);

- Accessible Max Address (see INCITS 558-2021 [B7]);
- Zone Activation (see INCITS 549-2021 [B5]);
- Storage Element Depopulation (see INCITS 558-2021 [B7]); and
- TCG locking ranges in a Locked state (see TCG Storage Interface Interactions Specification [B23]);

b) if verification is to be performed, write known data patterns to randomly selected logical blocks (see Clause 7 for recommendations on selection), to be used for verification after the *clear sanitization* method is complete;

c) perform one or more of the following actions:

- write commands (e.g., WRITE DMA EXT), with a fixed data value (e.g., all zeros) to all logical blocks from LBA = 0 to LBA = Native Max Address minus 1. Multiple passes or more complex values can be used;
- SCT Write Same, with a starting LBA = 0, and a length of 0;
- the SECURITY ERASE UNIT command in Normal Erase mode (see 8.4.2.3);
- for either:
  - TCG Pyrite SSC version 2.0 [B21] or later; or
  - TCG Opal SSC version 2.02 [B18] or later,

  if the Locking SP is owned (see TCG Storage Interface Interactions Specification [B23]), then invoke the following:
  - the Revert method on the AdminSP or the LockingSP; or
  - the RevertSP method on the AdminSP,

  with the ActiveDataRemovalMechanism column in the DataRemovalMechanism table set to:
  - Unmap;
  - Reset Write Pointers;
  - Block Erase; or
  - Overwrite Data Erase;

  and

d) if verification is to be performed, perform verification using the randomly selected blocks written earlier.

8.4.2.3 Clear by ATA Security Erase Unit in Normal Erase mode

If the SECURITY ERASE UNIT command is supported, and the security state is as follows:

- SEC4: Enabled/Locked/Not Frozen; or
- SEC5: Enabled/Not Locked/Not Frozen,

then the *host* shall do the following:

a) if verification is to be performed, write known data patterns to randomly selected logical blocks (see Clause 7 for recommendations on selection) to be used for verification after the *purge* actions are successful;

b) send the SECURITY ERASE UNIT command to the *device*, with the ERASE MODE bit cleared to 0 (i.e., Normal Erase mode); and

c) if verification is to be performed, perform verification using the randomly selected blocks written earlier.

If the SECURITY ERASE UNIT command does not complete successfully, then either retry until it completes successfully or use a different *clear sanitization* method (see 8.4.2.1). If no *clear sanitization* method is successful, then use the *purge sanitization* method (see 8.4.3) or the *destruct sanitization* method (see 8.4.4).

Given the variability in implementation of the SECURITY ERASE UNIT command, use of this command is not recommended without first consulting with the manufacturer to confirm that the *storage device* model-specific implementation meets the needs of the organization.

8.4.2.4 Clear for SCSI

To perform the *clear sanitization* method for SCSI, then the *host* shall, for each logical unit, do the following:

a) reset configuration options that limit the access to portions of the *storage media*, such as follows:

   1) the SCSI mode parameter block descriptor's NUMBER OF LOGICAL BLOCKS field (accessible with the MODE SENSE and MODE SELECT commands) (see INCITS 502-2019 [B3] and INCITS 506-2020 [B4]);

   2) Zone activation (see INCITS 549-2021 [B5]);

   3) Depopulation (see INCITS 506-2020 [B4] and INCITS 550 [B6]); and

   4) TCG Locking ranges (see TCG Storage Interface Interactions Specification [B23]) in a Locked state;

b) if verification is to be performed, write known data patterns to randomly selected logical blocks (see Clause 7 for recommendations on selection), to be used for verification after the *clear sanitization* method is complete;

c) perform one or more of the following actions:

   1) write commands (e.g., WRITE and WRITE SAME), with a fixed data value (e.g., all zeros) to all logical blocks from LBA = 0 to LB A = the highest numbered LBA. Multiple passes or more complex values can be used;

   2) the FORMAT UNIT command, with FFMT = 00b (i.e., the device server initializes the medium as specified in the CDB and parameter list before completing the format operation. After successful completion of the format operation, read commands and verify commands are processed as described in SBC-4);

   3) for either:

— TCG Pyrite SSC version 2.0 [B21] or later; or

— TCG Opal SSC version 2.02 [B18] or later,

and the Locking SP is owned (see TCG Storage Interface Interactions Specification [B23]), invoke one of the following:

— the Revert method on the AdminSP or the LockingSP; or

— the RevertSP method on the AdminSP,

with the ActiveDataRemovalMechanism column in the DataRemovalMechanism table set to:

— Unmap;

— Reset Write Pointers;

— Block Erase; or

— Overwrite Data Erase;

and

d) if verification is to be performed, perform verification using the randomly selected blocks written earlier.

8.4.2.5 Clear for NVMe

8.4.2.5.1 General requirements

The entire NVM subsystem is to be *cleared*.

An NVM subsystem consists of one or more controllers, as well as of zero or more namespaces. Not all namespaces are necessarily attached to each controller. Some namespaces are not necessarily attached to any controller. Some namespaces are shared between multiple controllers (i.e., attached to multiple controllers).

The *clear sanitization* method shall be applied to all namespaces in the NVM subsystem. If one or more namespaces are not attached to any controller and the *host* cannot attach such namespaces to a controller, then the *purge* (see 8.4.3) or the *destruct* (see 8.4.4) *sanitization* method shall be used instead.

The modifications to namespaces described in this subclause require that the namespace be attached to the controller performing the operation, and that the controller supports the I/O command set associated with that namespace. Because a namespace is not necessarily attached to a controller, performing these operations includes attaching each namespace to an appropriate controller.

If it is required to remove the data in one or more of the following:

— namespaces whose namespace write protection state is Permanent Write Protect;

— boot partitions;

— RPMBs; or

— authentication credentials for any of the following:

— NVMe in-band authentication;

— NVMe RPMB; or

— TCG Locking ranges in a Locked state,

and a technique (e.g., vendor specific) exists to remove the data, then use that technique; otherwise, the *destruct sanitization* method (see 8.4.4) shall be used.

If it is required to remove the data in any PMR, then the *purge sanitization* method (see 8.4.3) or the *destruct sanitization* method (see 8.4.4) shall be used instead.

Modification of write-protected namespaces can be configured to require authentication. Refer to the RPMB section of the NVMe Base Specification [B10].

The following technique is used to perform the *clear sanitization* method for NVMe:

a) *clear* all *user data* in namespaces, via one of the following:

— *clear* all namespaces with one command (see 8.4.2.5.2); or

— *clear* each namespace individually (see 8.4.2.5.3);

and

b) *clear* all *user data* associated with controllers (see 8.4.2.5.4).

8.4.2.5.2 Clear all namespaces with one command

If any namespace is associated with the Key Value command set, then the clear each namespace individually technique (see 8.4.2.5.3) shall be used instead of this technique.

If the Format NVM command does not support the Namespace ID field set to FFFF_FFFFh [i.e., bit 3 of the Format NVM Attributes field in the I/O Command Set Independent Identify Controller data structure is set to 1b], then the clear each namespace individually technique (see 8.4.2.5.3) shall be used instead of this technique.

The following technique is used to *clear* all namespaces with one command issued to a controller:

a) reset configuration options that limit access to portions of the *storage media*, such as follows:

1) attach all namespaces to that controller; and

2) for all namespaces attached to that controller, change the namespace write protection state to No Write Protect.

If not all namespaces can be attached to that controller, or there was a failure to remove write protection from a write-protected namespace, then the clear each namespace individually technique (see 8.4.2.5.3) shall be used instead of this technique;

b) if verification is to be performed, write known data patterns to randomly selected[7] logical blocks (see Clause 7 for recommendations on selection), to be used for verification after the *clear sanitization* method is complete;

c) *Clear* all *user data* in all namespaces:

— perform the Format NVM command, with the Namespace ID field set to FFFF_FFFFh (i.e., all namespaces in the NVM subsystem) on any controller in the NVM subsystem; or

— if either:

— TCG Pyrite SSC version 2.0 [B21] or later; or

— TCG Opal SSC version 2.02 [B18] or later,

are supported and the Locking SP is owned (see TCG Storage Interface Interactions Specification [B23]), then invoke:

— the Revert method on the AdminSP or the LockingSP; or

— the RevertSP method on the AdminSP,

with the ActiveDataRemovalMechanism column in the DataRemovalMechanism table set to the following:

— Unmap;

— Reset Write Pointers;

— Block Erase, or

— Overwrite Data Erase;

and

d) if verification is to be performed, perform verification using the randomly selected blocks written earlier.

8.4.2.5.3 Clear each namespace individually

The following technique is used to *clear* all namespaces individually:

a) reset configuration options that limit access to portions of the *storage media*, such as follows:

1) attach all namespaces to at least one controller; and

2) for all attached namespaces, change the namespace write protection state to No Write Protect.

If not all namespaces can be attached, or there was a failure to remove write protection from a write-protected namespace, then use a different technique to *clear user data* from namespaces (see 8.4.2.5.1);

b) for all attached namespaces using the NVMe Command Set Specification [B12] or the NVMe Zoned Namespace Command Set Specification [B13]:

[7] The ZNS Command Set requires additional blocks to be written to write a randomly selected block not at the start of a zone.

1) if verification is to be performed, write known data patterns to randomly selected[8] logical blocks (see Clause 6 for recommendations on selection), to be used for verification after the *clear sanitization* method is complete;
2) write commands (e.g., Write, Format NVM) with a fixed data value (e.g., all zeros) to all logical blocks from LBA = 0 to the highest numbered LBA on this namespace. Multiple passes or more complex values can be used; and
3) if verification is to be performed, perform verification using the randomly selected blocks written earlier;

and

c) for all attached namespaces using the NVMe Key Value Command Set Specification [B11]:
   1) if verification is to be performed, write known data patterns to randomly selected Key-value keys (see Clause 7 for recommendations on selection), to be used for verification after the *clear sanitization* method is complete;
   2) delete all key/value pairs from the namespace; and
   3) if verification is to be performed, verify that no key-value keys exist in the namespace.

8.4.2.5.4 Clear all user data associated with controllers

The following technique shall be used to *clear* all *user data* associated with controllers.

The *host* shall, for each controller, do the following:

a) for each CMB, if any:
   1) delete any I/O Submission Queues and Completion Queues in the CMB;
   2) if verification is to be performed, write known data patterns to randomly selected address ranges (see Clause 7 for recommendations on selection), for verification after the next operation is complete;
   3) write a fixed data value (e.g., all zeros) to all address. Multiple passes or more complex values can be used; and
   4) if verification is to be performed, perform verification using the randomly selected address ranges that were written.

b) for each HMB:
   1) disable the HMB by using a Set Features command to clear the EHM bit to 0b in the HMB Feature (Feature Identifier 0Dh) without releasing HMB memory to the host software for reuse;
   2) if verification is to be performed, write known data patterns to randomly selected address ranges (see Clause 7 for recommendations on selection), for verification after the next operation is complete;
   3) write a fixed data value (e.g., all zeros) to all addresses. Multiple passes or more complex values can be used; and
   4) if verification is to be performed, perform verification using the randomly selected address ranges that were written.

[8] The ZNS Command Set requires additional blocks to be written to write a randomly selected block not at the start of a zone.

8.4.3 Purge

8.4.3.1 General

The *purge sanitization* method consists of performing the operations specified in the following subclauses:

— 8.4.3.2 for ATA *storage devices*;

— 8.4.3.3 for SCSI *storage devices*; and

— 8.4.3.4 for NVMe subsystems.

In addition to the operations above, the *purge sanitization* method can also include vendor-specific operations.

If for any element of a *storage device*, any operation in the *purge sanitization* method fails, or if verification fails, then the *sanitization* method has failed. The *purge sanitization* method can be retried until successful, or the *destruct sanitization* method (see 8.4.4) can be used.

Degaussing (see 6.5.4) of HDD *storage devices* generally results in rendering the *storage device* permanently inoperable.

8.4.3.2 Purge for ATA

To perform the *purge sanitization* method for ATA *storage devices*, then the *host* shall do the following:

a) if verification is to be performed, write known data patterns to randomly selected logical blocks (see Clause 7 for recommendations on selection) to be used for verification after the *purge* actions are successful;

b) perform one or more of the following actions:

   1) *cryptographic erase* (see 8.4.3.5);

   2) sanitize block erase (see 8.4.3.6);

   3) sanitize overwrite (see 8.4.3.7); or

   4) the SECURITY ERASE UNIT command in Enhanced Erase mode (see 8.4.3.2.1).

c) if none of these methods is supported, use a *destruct sanitization* method (see 8.4.4) instead of a *purge sanitization* method;

d) optionally, perform one invocation of the *clear sanitization* method (see 8.4.2) on the *storage media*;

   and

e) if verification is to be performed, perform verification using the randomly selected blocks written earlier.

8.4.3.2.1 Purge by ATA Security Erase Unit in Enhanced Erase mode

If the SECURITY ERASE UNIT command is supported, and the security state is as follows:

— SEC4: Enabled/Locked/Not Frozen, or

— SEC5: Enabled/Not Locked/Not Frozen.

then the *host* shall do the following:

a) if verification is to be performed, write known data patterns to randomly selected logical blocks (see Clause 7 for recommendations on selection) to be used for verification after the *purge* actions are successful;

b) send the SECURITY ERASE UNIT command to the *device*, with the ERASE MODE field set to 1 (i.e., Enhanced Erase mode); and

c) if verification is to be performed, perform verification using the randomly selected blocks written earlier.

If the SECURITY ERASE UNIT command does not complete successfully, then either retry until it completes successfully or use a different *purge sanitization* method (see 8.4.3.1). If no *purge sanitization* method is successful, then use the *destruct sanitization* method (see 8.4.4).

Given the variability in implementation of the SECURITY ERASE UNIT command, use of this command is not recommended without first consulting with the manufacturer to confirm that the *storage device* model-specific implementation meets the needs of the organization.

8.4.3.3 Purge for SCSI

To perform the *purge sanitization* method for SCSI *storage devices*, then the *host* shall do the following:

a) if verification is to be performed, write known data patterns to randomly selected logical blocks (see Clause 7 for recommendations on selection) to be used for verification after the *purge* actions are successful;

b) perform one or more of the following actions:

— *cryptographic erase* (see 8.4.3.5);

— sanitize block erase (see 8.4.3.6); and

— sanitize overwrite (see 8.4.3.7);

c) if none of these *sanitization* methods is supported, use the *destruct sanitization* method (see 8.4.4);

d) optionally, perform one invocation of the *clear sanitization* method (see 8.4.2) on the *storage media*;

and

e) if verification is to be performed, perform verification using the randomly selected blocks written earlier.

8.4.3.4 Purge for NVMe

8.4.3.4.1 General requirements

If it is required to remove the data in:

- namespaces whose namespace write protection state is Permanent Write Protect;
- boot partitions;
- divided domains;
- RPMBs; and
- authentication credentials for any of the following:
  - NVMe In-Band Authentication; and
  - NVMe RPMB,

and a technique (e.g., vendor specific) exists to remove the data, then use that technique; otherwise, the *destruct sanitization* method (see 8.4.4) shall be used.

If verification of *sanitization* of namespaces associated with the NVMe Key Value Command Set Specification [B11] is required, the *destruct sanitization* method (see 8.4.4) shall be used instead.

8.4.3.4.2 Specific requirements

To perform the *purge sanitization* method for NVM subsystems, the *host* shall do the following:

a) reset configuration options that limit the access to portions of the *storage media*:
   - change the namespace write protection state to No Write Protect for all namespaces;
   - disable each HMB regions by using a Set Features command to clear the EHM bit to 0b in the HMB Feature (Feature Identifier 0Dh) without releasing the HMB memory to the host software for reuse; and
   - disable the PMR[9] (if supported and enabled);

   if any of these resulted in a failure, then the *destruct sanitization* method (see 8.4.4) shall be used instead;

b) delete all I/O submission and completion queues by issuing the Delete I/O Submission Queue command and the Delete I/O Completion Queue command;

c) if verification is to be performed, then:
   1) for each namespace associated with a logical block-based I/O command set (e.g., the NVM Command set), write known data patterns to randomly selected[10] logical blocks (see Clause 7 for recommendations on selection criteria), to be used for verification after the *purge* actions are successful;

[9] NVM subsystems that implement a PMR require additional considerations. TCG *cryptographic erase* methods on such NVM subsystems are not defined to *purge* PMR data and shall not be used to *purge* such NVM subsystems. The NVMe Sanitize command is defined to *purge* PMR data.
[10] The ZNS Command Set requires additional blocks to be written to write a randomly selected block not at the start of a zone.

2) for each CMB (if any), write known data patterns to randomly selected address ranges (see Clause 7 for recommendations on selection), to be used for verification after the *purge sanitization* method is complete:

3) for each HMB (if any), write known data patterns to randomly selected address ranges (see Clause 7 for recommendations on selection), to be used for verification after the *clear sanitization* method is complete;

d) successfully perform one or more of the following actions:

1) *cryptographic erase* (see 8.4.3.5);

2) sanitize block erase (see 8.4.3.6); or

3) sanitize overwrite (see 8.4.3.7);

e) for each CMB (if any), write zeros to all addresses in the CMB;

f) for each HMB (if any), write zeros to all addresses in the HMB;

and

g) if the actions performed in step d) through step f) succeed and verification is to be performed, perform verification:

1) for each namespace associated with a logical block-based I/O command set (e.g., the NVM Command Set), perform verification using the randomly selected blocks copied in step c);

2) for each CMB, perform verification using the randomly selected address ranges that were written in step c); and

3) for each HMB, perform verification using the randomly selected address ranges that were written in step c).

In addition to the operations above, the *purge sanitization* method can also include vendor-specific operations.

If, for any element of a *storage device*, any operation in the *purge sanitization* method fails, or if verification fails, then the *sanitization* method has failed. The *purge sanitization* method (see 8.4.3.1) can be retried until successful, or the *destruct sanitization* method (see 8.4.4) can be used.

8.4.3.5 Purge by cryptographic erase

Do not use this *sanitization* method if the following occurs:

— the *storage device* does not support encryption (e.g., TCG Pyrite SSC [B20]);

— none of the techniques specified in this clause are supported;

— the encryption was enabled after sensitive data were stored on the *storage device*;

— some data are encrypted, and some data are not encrypted; or

— any NVMe namespaces are not encrypted.

Not all implementations of encryption are necessarily suitable for reliance on *cryptographic erase* as a *purge* mechanism. The decision regarding whether to use *cryptographic erase* depends on verification of attributes previously identified in this guidance and in Annex B. Issue commands as necessary to cause all MEKs to be changed (if the technical specifications described in this standard have been satisfied). Refer to TCG specifications and *storage device* manufacturers for more information.

Cryptographic erase is done using one of the following techniques by the *host*:

- if the *storage device* supports the TCG Opal SSC version 2.02 [B18] or later, and the Locking SP is owned (see TCG Storage Interface Interactions Specification [B23]), then:
  - invoke the Revert method on the AdminSP or the LockingSP; or
  - invoke the RevertSP method on the AdminSP,

  with the ActiveDataRemovalMechanism column in the DataRemovalMechanism table set to Crypto Erase.
- if the *storage device* supports:
  - TCG Opal SSC (prior to version 2.02) [B17];
  - TCG Opalite SSC [B19]; or
  - TCG Ruby SSC [B22],

  and the Locking SP is owned (see TCG Storage Interface Interactions Specification [B23]), then:
  - invoke the Revert method on the AdminSP or the LockingSP; or
  - invoke the RevertSP method on the AdminSP;
- if the *storage device* supports the TCG Enterprise SSC and the Locking SP is owned, then invoke the Erase method on all locking ranges; or
- if the *storage device* supports a Sanitize cryptographic command, then use the appropriate command in Table 9.

NVM subsystems that implement PMR require additional considerations. TCG *cryptographic erase* methods on such NVM subsystems are not defined to purge PMR data, and they shall not be used to purge such NVM subsystems. The NVMe Sanitize command is defined to purge PMR data.

Table 9—Sanitize cryptographic erase commands

Interface	Command
ATA	CRYPTO SCRAMBLE EXT
SCSI	SANITIZE, with the SERVICE ACTION field set to 03h (i.e., CRYPTOGRAPHIC ERASE)
NVMe	Sanitize, with the Sanitize Action field set to 110b (i.e., Start a Crypto Erase Sanitize operation). If vendor documentation asserts that Format NVM with the Secure Erase Settings (SES) field set to Cryptographic Erase (i.e., 010b) meets the requirements of *purge* in this standard, then Format NVM with the SES field set to Cryptographic Erase (i.e., 010b) is useable as a *purge sanitization* method.

If the selected *cryptographic erasure* method does not complete successfully, then either retry until it completes successfully or use a different *purge sanitization* method (see 8.4.3.1).

If no *purge sanitization* method is successful, then use the *destruct sanitization* method (see 8.4.4).

8.4.3.6 Purge by sanitize block erase

If the *storage device* supports any TCG Opal Family SSC (e.g., Opal [B17], [B18]; Opalite [B19]; Pyrite [B20], [B21]; and Ruby [B22]) or TCG Enterprise SSC [B16], and the Locking SP is owned (see TCG Storage Interface Interactions Specification [B23]), then use a different *purge sanitization* method (see 8.4.3). These TCG methods do not specify *sanitization* of *user data* by block erase methods.

If the *storage device* supports a Sanitize Block Erase command, then use the appropriate command in Table 10.

The ATA, SCSI, and NVMe requirements for *storage devices* that implement the commands in Table 10 are to make any storage media no longer alterable (e.g., bad blocks) no longer accessible across the *host interface*. If this functionality poses unacceptable risk to the organization, then the *destruct sanitization* method (see 8.4.4) should be used instead of the *purge sanitization* method (see 8.4.3.1).

Table 10—Sanitize block erase commands

Interface	Command
ATA	BLOCK ERASE EXT
SCSI	SANITIZE, with the SERVICE ACTION field set to 02h (i.e., BLOCK ERASE)
NVMe	Sanitize, with the Sanitize Action field set to 010b (i.e., Start a Block Erase Sanitize operation). If verification (see Clause 7) is to be performed, then the No-Deallocate After Sanitize bit in the Sanitize command shall be set to 1. If verification is to be performed and the No-Deallocate Inhibited bit is set to 1 in the I/O Command Set Independent Identify Controller data structure, and: — The *sanitize* operation completes successfully with deallocation of all *user data* (i.e., bits 2:0 of the Sanitize Status field in the Sanitize Status log [age are set to 100b), then a *device* format that writes all *addressable storage media* is required before verification. — The *sanitize* operation completes successfully without deallocation of all *user data* (i.e., bits 2:0 of the Sanitize Status field in the Sanitize Status log page are set to 001b), then the NVM subsystem is ready for verification. — The Sanitize command fails or the *sanitize* operation fails, then the selected block erasure technique did not complete successfully.

If the selected block erasure technique does not complete successfully, then either retry until it completes successfully or use a different *purge sanitization* method (see 8.4.3).

If no *purge sanitization* method is successful, then use the *destruct sanitization* method (see 8.4.4).

8.4.3.7 Purge by sanitize overwrite

If the *storage device* supports any TCG Opal Family SSC (e.g., Opal [B17], [B18]; Opalite [B19]; Pyrite [B20], [B21]; and Ruby [B22]) or TCG Enterprise SSC [B16], and the Locking SP is owned, then use a different *purge sanitization* method (see 8.4.3.1). These TCG methods do not specify *sanitization* of *user data* by overwrite methods.

The endurance of some types of *storage media* (e.g., NAND) is adversely affected by overwriting. *Storage devices* that use that type of *storage media* should *purge* by Sanitize Block Erase instead (see 8.4.3.6).

If the *storage device* supports a Sanitize Overwrite command, then use the appropriate command in Table 11 to do the following:

— apply one pass of a fixed pattern (e.g., all zeros or a pseudo-ramdom value) across the *storage media* surface;

— apply more than one pass with a fixed pattern; or

— apply an odd number of passes, leveraging the invert option so that every other pass is the inverted version of the pattern specified.

The ATA, SCSI, and NVMe requirements for *storage devices* that implement the commands in Table 11 are to make any *storage media* no longer alterable (e.g., bad blocks) inaccessible across the *host interface*. If this functionality poses unacceptable risk to the organization, then the *destruct sanitization* method (see 8.4.4) should be used instead of the *purge sanitization* method (see 8.4.3.1).

Table 11—Sanitize overwrite commands

Interface	Command
ATA	OVERWRITE EXT
SCSI	SANITIZE, with the SERVICE ACTION field set to 01h (i.e., OVERWRITE)
NVMe	Sanitize, with the Sanitize Action field set to 011b (i.e., Start an Overwrite Sanitize operation). If verification (see Clause 7) is to be performed, then the No-Deallocate After Sanitize bit in the Sanitize command shall be set to 1. If verification is to be performed and the No-Deallocate Inhibited bit is set to 1 in the I/O Command Set Independent Identify Controller data structure, and: — the *sanitize* operation completes successfully with deallocation of all *user data* (i.e., bits 2:0 of the Sanitize Status field in the Sanitize Status log page are set to 100b), then a *device* format that writes all *addressable storage media* is required before verification; — the *sanitize* operation completes successfully without deallocation of all *user data* (i.e., bits 2:0 of the Sanitize Status field in the Sanitize Status log page are set to 001b), then the NVM subsystem is ready for verification; and — the Sanitize command fails or the *sanitize* operation fails, then the selected overwrite technique did not complete successfully.

If the selected overwrite technique does not complete successfully, then either retry until it completes successfully or use a different *purge sanitization* method (see 8.4.3.1).

If no *purge sanitization* method is successful, then use the *destruct sanitization* method (see 8.4.4).

8.4.4 Destruct

The *destruct sanitization* method makes recovery of any *target data* infeasible and leaves the *storage device* nonoperational for any use.

The *destruct sanitization* method shall consist of the following:

a) performing one or more of the following techniques:

— *incinerate* by burning the *storage device* in a licensed incinerator;

— *melt* by changing *storage media* from a solid to a liquid state; or

— *disintegrate*;

and

b) if verification is to be performed, then perform verification as described in Clause 7.

8.5 Other magnetic media

8.5.1 Floppy disk

Table 12 summarizes where to find the specifications for *clear*, *purge*, and *destruct* for floppy disks.

Table 12—Floppy disk sanitization

Sanitization method	Reference
Clear	8.5.1.1
Purge	8.5.1.2
Destruct	8.5.1.3

8.5.1.1 Clear

The *storage media* should be overwritten by using organizationally approved software, and the data should be verified to organizationally approved levels (see Clause 7).

The *clear sanitization* method consists of at least a single pass of writes with a fixed data value (e.g., all zeroes). Multiple passes or more complex values may be used.

8.5.1.2 Purge

The *storage media* should be *degaussed* in an organizationally approved degausser. See 6.5.4.

8.5.1.3 Destruct

See 6.4.

8.5.2 Removable flexible or rigid magnetic disks

These disks include Zip, Floptical, Jaz, SyQuest, LS-120, and so on (Table 13).

Table 13—Removable flexible or rigid magnetic disk sanitization

Sanitization method	Reference
Clear	8.5.2.1
Purge	8.5.2.2
Destruct	8.5.2.3

8.5.2.1 Clear

The *storage media* should be overwritten by using organizationally approved software, and the data should be verified to organizationally approved levels (see Clause 7).

The *clear sanitization* method consists of at least a single pass of writes with a fixed data value (e.g., all zeroes). Multiple passes or more complex values may be used.

8.5.2.2 Purge

The *storage media* should be *degaussed* in an organizationally approved degausser. See 6.5.4.

8.5.2.3 Destruct

See 6.4.

8.5.3 Reel and cassette format magnetic tapes

These tapes include LTO, 8 mm, DDS, DAT, DLT, QIC, and so on (Table 14).

Table 14—Reel and cassette format magnetic tape sanitization

Sanitization method	Reference
Clear	8.5.3.1
Purge	8.5.3.2
Destruct	8.5.3.3

8.5.3.1 Clear

Re-record (overwrite) all data on the tape using an organizationally approved pattern (e.g., video noise) using a system with similar characteristics to the one that originally recorded the data. For example, overwrite previously recorded VHS format video signals on a comparable VHS format recorder. All portions of the magnetic tape should be overwritten one time with an organizationally approved pattern. *Clearing* a magnetic tape by re-recording (overwriting) it can be impractical for most applications because the technique occupies the tape transport for excessive time periods.

If the tape is of the WORM type, then it cannot be erased or overwritten; in that case, use the *purge sanitization* method (see 8.5.3.2).

8.5.3.2 Purge

The *storage media* should be degaussed in an organizationally approved degausser. Magnetic *storage media* exists for which *degaussing* is not a valid *purge sanitization* method. See 6.5.4.

If the *storage media* cannot be effectively purged, then the *destruct sanitization* method (see 8.5.3.3) shall be used instead.

If the tape is in a cartridge containing a MAM, then use a vendor-specific application to reset the MAM contents to factory default.

8.5.3.3 Destruct

The *storage media* should be *melted* by changing *storage media* from a solid to a liquid state.

Preparatory steps for *destruct* (e.g., removing a tape from the reel or a cassette prior to destruction) are unnecessary. However, segregation of components (tape and reels or cassettes) is possibly necessary to comply with the requirements of a destruction facility or for recycling measures.

See 6.4.

8.6 USB removable media

These media include Pen Drives, Thumb Drives, Flash Drives, Memory Sticks, and so on.

USB *storage devices* are SSD *devices* that use the SCSI command *host interface*. See 8.4.

8.7 Memory cards

These cards includes SD, SDHC, MMC, Compact Flash, Microdrive, MemoryStick, and so on (Table 15).

Table 15—Memory cards sanitization

Sanitization method	Reference
Clear	8.7.1
Purge	8.7.2
Destruct	8.7.3

8.7.1 Clear

The *storage media* should be overwritten by using organizationally approved software, and the data should be verified to organizationally approved levels (see Clause 7).

The *clear sanitization* method consists of at least two passes of writes, to include a pattern in the first pass and its complement in the second pass. Additional passes can be used.

Optional protocol-specific commands can perform a *clear sanitization* method that can be organizationally approved.

8.7.2 Purge

Not applicable. See *destruct* (8.7.3).

Optional protocol-specific commands can perform a *purge sanitization* method that can be organizationally approved.

8.7.3 Destruct

See 6.4.

8.8 Embedded flash on boards and storage devices

These devices include motherboards and peripheral cards (e.g., network adapters or any other adapter containing nonvolatile flash memory; Table 16).

Table 16—Embedded flash on boards and storage devices sanitization

Sanitization method	Reference
Clear	8.8.1
Purge	8.8.2
Destruct	8.8.3

8.8.1 Clear

If supported by the *storage device*, then the state should be reset to original factory settings.

8.8.2 Purge

Not applicable. See *destruct* (8.8.3).

The electronics boards from the system should be destructed. See *destruct* (8.8.3).

8.8.3 Destruct

See 6.4.

8.9 RAM and ROM-based storage devices

8.9.1 DRAM

Table 17 summarizes where to find the specifications for clear, purge, and destruct for DRAM.

Table 17—DRAM sanitization

Sanitization method	Reference
Clear	8.9.1.1
Purge	8.9.1.2
Destruct	8.9.1.3

8.9.1.1 Clear

See *purge* (8.9.1.2).

8.9.1.2 Purge

A *storage device* containing DRAM should be powered off and removed from the power source, and the battery should be removed (if battery backed). Alternatively, the DRAM should be removed from the *storage device*.

In either case, the DRAM should remain without power for a period of at least 5 min.

8.9.1.3 Destruct

See 6.4.

8.9.2 EAPROM

Table 18 summarizes where to find the specifications for clear, purge, and destruct for EAPROM.

Table 18—EAPROM sanitization

Sanitization method	Reference
Clear	8.9.2.1
Purge	8.9.2.2
Destruct	8.9.2.3

8.9.2.1 Clear

See *purge* (8.9.2.2).

8.9.2.2 Purge

A full-chip *purge* should be performed as per the manufacturer's data sheets.

8.9.2.3 Destruct

See 6.4.

8.9.3 EEPROM

Table 19 summarizes where to find the specifications for *clear*, *purge*, and *destruct* for EEPROM.

Table 19—EEPROM sanitization

Sanitization method	Reference
Clear	8.9.3.1
Purge	8.9.3.2
Destruct	8.9.3.3

8.9.3.1 Clear

See *purge* (8.9.3.2).

8.9.3.2 Purge

The *storage media* should be overwritten by using organizationally approved techniques, and the data should be verified to organizationally approved levels (see Clause 7).

8.9.3.3 Destruct

Some *storage media* types are not specifically addressed by this standard, but the processes described in this standard guide *storage media sanitization* decision making regardless of the type of *storage media* in use.

See 6.4.

Annex A

(normative)

Storage devices with embedded storage

Products may contain multiple *storage devices*, and the end users of those products can be unable to directly perform the *sanitization* methods described in this standard. Examples include the following:

- a mobile phone can contain personal messages, photos, browsing histories, or navigation data stored on one or more SSDs soldered to a circuit board;
- a copier/printer can contain images of documents and user credentials stored on an HDD or SSD;
- a network router can contain network configurations and administrator credentials;
- an automobile can contain navigation data, user credentials, browsing histories, telephone records, and copies of licensed movies; and
- a digital television can contain user credentials, browsing histories, and copies of licensed movies.

A common feature of these products is that it is not practical for the end user to remove and *sanitize* the embedded *storage devices*. Instead, the product designer provides *sanitization* capabilities in the user interface. The user options can be 1) separate clearing of different types of data and 2) a "factory reset" that *sanitizes* all data.

For such products, the designer defines *sanitization* operations meaningful to the end user and easy to invoke. The product designer then uses this standard to decompose those user-level operations into invocations of *sanitization* methods for each *storage device* embedded in the product.

Disassembly of battery and display can be required.

Refer to the manufacturer for additional information on the proper *sanitization* procedure, as well as for details about implementation differences between *storage device* versions and operating system versions. Proper initial configuration using guides helps ensure that the level of data protection and *sanitization* assurance is as robust as possible. A defined *sanitization* procedure may or may not be available.

If the *storage device* contains removable or embedded *storage media*, ensure that the *storage media* is *sanitized* using appropriate *storage-media*-dependent procedures specified in Clause 8.

Following a successful *clear* or *purge* operation, manually navigate to multiple areas of the *storage device* (e.g., call history, passwords, browser history, files, or photos) to verify that no personal information has been retained on the *storage device*.

A.1 Networking device

A.1.1 Routers, hubs, and switches (home, home office, enterprise)

Table A.1 summarizes where to find the specifications for *clear*, *purge*, and *destruct* for this *device* type.

Table A.1—Router and switch sanitization

Sanitization method	Reference
Clear	A.1.1.1
Purge	A.1.1.2
Destruct	A.1.1.3

Network *devices* can contain removable or embedded *storage media* or *storage devices*. The *storage media* should be removed and *sanitized* using *storage media* techniques.

A.1.1.1 Clear

Perform a full manufacturer's reset to return the router or switch back to its factory default settings.

Refer to the manufacturer for additional information on the proper procedure.

A.1.1.2 Purge

Most hubs, routers, and switches only offer capabilities to *clear* (and not *purge*) the data contents. A router or switch can offer *purge* capabilities, but these capabilities are specific to the hardware and firmware of the *device* and should be applied with caution. Refer to the *device* manufacturer to identify whether the *device* has a *purge* capability that applies *storage-media*-dependent techniques (e.g., overwriting or block erasing) to help ensure that data recovery is infeasible, and that the *device* does not simply remove the file pointers.

Refer to the manufacturer for additional information on the proper *sanitization* procedure.

See *destruct* (A.1.1.3).

A.1.1.3 Destruct

See 6.4.

A.2 Equipment

A.2.1 Office equipment

This equipment includes copy, print, fax, and multifunction machines.

Office equipment can contain removable or embedded *storage media*. Removable *storage media* should be removed and *sanitized* using *storage media* techniques.

For both the *clear sanitization* method and (if applicable) the *purge sanitization* method, use the user interface to navigate to multiple areas of the *device* (e.g., stored fax numbers and network configuration information) to verify that no personal information has been retained on the *device*.

For both the *clear sanitization* method and (if applicable) the *purge sanitization* method, the ink, toner, and associated supplies (drum, fuser, etc.) should be removed and destroyed or disposed of in accordance with applicable law, environmental, and health considerations. Some of these supplies may retain impressions of

data printed by the machine and therefore could pose a risk of data exposure and should be handled accordingly.

If the *device* is functional, one way to reduce the associated risk is to print a blank page, then an all-black page, and then another blank page.

For *devices* with dedicated color components (e.g., cyan, magenta, and yellow toners and related supplies), one page of each color should also be printed between blank pages. The resulting sheets should be handled at the confidentiality of the office equipment (prior to *sanitization*).

Note that these procedures do not apply to supplies (e.g., as ink/toner on a one-time use roll) as they are generally not used again and therefore are not addressed by sending additional pages through the equipment. Office equipment supplies can also pose health risks and should be handled using appropriate procedures to reduce exposure to the print components and toner.

For both the *clear sanitization* method and (if applicable) the *purge sanitization* method, refer to the manufacturer for additional information on the proper *sanitization* procedure.

Table A.2 summarizes where to find the specifications for *clear*, *purge*, and *destruct* for this *device* type.

Table A.2—Office equipment sanitization

Sanitization method	Reference
Clear	A.2.1.1
Purge	A.2.1.2
Destruct	A.2.1.3

A.2.1.1 Clear

Perform a full manufacturer's reset to return the office equipment to its factory default settings.

A.2.1.2 Purge

See *destruct* (A.2.1.3).

Most office equipment only offers capabilities to *clear* (and not *purge*) the data contents. Office equipment can offer *purge* capabilities, but these capabilities are specific to the hardware and firmware of the *device* and should be applied with caution. Refer to the *device* manufacturer to identify whether the *device* has a *purge* capability that applies *storage-media*-dependent techniques (e.g., overwriting or block erasing) or *cryptographic erase* to help ensure that data recovery is infeasible, and that the *device* does not simply remove the file pointers.

Office equipment can have removable *storage media*, and if so, *storage-media*-dependent *sanitization* techniques can be applied to the associated *storage device*.

A.2.1.3 Destruct

See 6.4.

A.3 Devices with built-in storage

These devices include phones, tablets, *media* players, watches, game consoles, and so on.

Table A.3 summarizes where to find the specifications for *clear*, *purge*, and *destruct* for this *device* type.

Table A.3—Devices with built-in storage

Sanitization method	Reference
Clear	A.3.1
Purge	A.3.2
Destruct	A.3.3

A.3.1 Clear

Use an organizationally approved method (e.g., factory reset) that removes access to *user data*. Refer to the manufacturer for additional information on the proper *sanitization* procedure, as well as for details about implementation differences between *device* versions and operating system versions. Proper initial configuration using guides helps ensure that the level of data protection and sanitization assurance is as robust as possible. If the *device* contains removable *storage media*, ensure that the *storage media* is *sanitized* using appropriate *storage-media*-dependent procedures.

If an organizationally approved *sanitization* method does not exist, then use *purge* (see A.3.2).

A.3.2 Purge

Use an organizationally approved technique (e.g., factory reset) that deletes the *user data*. Refer to the manufacturer for additional information on the proper *sanitization* procedure, as well as for details about implementation differences between *device* versions and operating system versions. Proper initial configuration using guides helps ensure that the level of data protection and *sanitization* assurance is as robust as possible. If the *device* contains removable *storage media*, ensure that the *storage media* is *sanitized* using appropriate *storage-media*-dependent procedures.

Many *devices* with built-in *storage media* only offer capabilities to *clear* (and not *purge*) the data contents. A *storage device* with built-in *storage media* can offer *purge* capabilities, but these capabilities are specific to the hardware and software of the *storage device* and should be applied with caution.

The *device* manufacturer or service provider can provide a vendor-specific *purge* capability that applies organizationally acceptable *storage-media*-dependent techniques (e.g., overwriting or block erasing) or *cryptographic erase* to help ensure that data recovery is infeasible. This vendor-specific *purge sanitization* method should be organizationally reviewed for *purge* effectiveness (e.g., review if the *device* manufacturer *purge sanitization* method retains system or network information).

If the effectiveness of the *purge sanitization* method is not acceptable, then use the *destruct sanitization* method (see A.3.3).

A.3.3 Destruct

See 6.4.

Annex B

(informative)

Cryptographic erase

Cryptographic erase can provide significant benefits in both timeliness and assurance and is widely available from nearly all major *storage device* vendors. *Cryptographic erase* could provide substantial value by doing the following:

- facilitating rapid eradication of sensitive data (in seconds versus hours or days);
- reducing the wear on the *storage device* (therefore potentially extending the life of the *storage device*);
- reducing the amount of time expended performing *sanitization*;
- making it easier to safely repurpose *storage devices*, instead of destroying them;
- using only a well vetted cryptographic implementation to avoid potential for errors in implementation or use of weak cryptographic algorithms; and
- addressing *storage media* types that are impractical to address using legacy *degaussing* and destruction techniques.

Cryptographic erase leverages the encryption of *target data* by enabling *sanitization* of the *target data*'s encryption key. This leaves only the ciphertext remaining on the *storage media*, effectively sanitizing the data.

Without the encryption key used to encrypt the *target data*, the data are unrecoverable. The level of effort needed to decrypt this information without the encryption key then is the lesser of either of the following:

- the strength of the cryptographic algorithm used to encrypt the data (including mode of operation); and
- the level of entropy of the cryptographic key generation algorithm.

As a result, *sanitization* of the data is reduced to *sanitization* of the encryption key(s) used to encrypt the data. With *cryptographic erase*, *sanitization* can be performed with high assurance much faster than with other *sanitization* techniques. The encryption itself acts to *sanitize* the data.

Generally, *cryptographic erase* can be executed in seconds. This is especially important as *storage devices* get larger resulting in other *sanitization* methods taking more time. *Cryptographic erase* can also be used as a supplement or in addition to other *sanitization* approaches.

Reliance on *cryptographic erase* to *purge* the *storage media* on *storage devices* is not appropriate if the following is true:

- the encryption was enabled after sensitive data were stored on the *storage device* without having been *sanitized* first; or
- it is unknown whether sensitive data were stored on the *storage device* without being *sanitized* prior to encryption.

Whereas *cryptographic erase* is intended for use to *purge* the *storage media* (including SEDs, mobile *storage devices*, and other *storage devices*), the level of assurance depends on the following:

— whether or not any unencrypted *user data* were stored on the *storage device* prior to encrypting new *user data* written to the *storage device*, and whether that previously unencrypted *user data* were subsequently encrypted (e.g., a mixture of unencrypted and encrypted *user data*);

— locations in the *storage device* where the data encryption key is stored (be it the *target data*'s encryption key or a wrapping key that encrypts the *target data*'s encryption key);

— all copies of the encryption keys used to encrypt the *target data* are *sanitized*;

— if the *target data*'s encryption keys are, themselves, encrypted with one or more wrapping keys, it is acceptable to perform *cryptographic erase* by *sanitizing* the wrapping key(s) necessary to prevent decryption of the data encryption key(s); and

— the ability of a user to clearly identify the commands provided by the *storage device* to perform the *cryptographic erase* operation.

Other *cryptographic erase* considerations include the following:

— if the encryption key (or any other key intended to become no longer retrievable as a consequence of *cryptographic erase sanitization* of another key) exists outside of the *storage device* (generally due to escrow or injection), a possibility exists that the key could be used in the future to recover data stored on the encrypted *storage media*; and

— all copies of encryption keys (including those escrowed or stored in a key management appliance) need to be *sanitized* for *cryptographic erase* to be successful.

The choice regarding whether to leverage *cryptographic erase* on a given *storage device* depends on the organizational requirements for *sanitization*, as well as potentially on the end user's ability to determine whether the implementation offers sufficient assurance against future recovery of the data. The level of assurance depends in large part on the factors described in Table B.1.

Table B.1—Cryptographic erase considerations

Area	Consideration(s)
Key generation	The level of entropy of the random number sources and quality of key generation procedures applied to the random data. This applies to the cryptographic keys and to the wrapping keys (if any) affected by the *cryptographic erase* operation.
Media encryption	The security strength and validity of implementation of the encryption algorithm/mode used for protection of the *target data*.
Key wrapping	The key being *sanitized* might not be the MEK, but instead a key used to wrap (that is, encrypt) the MEK or another key. In this case, the security strength and level of assurance of the wrapping techniques used are advised to be commensurate with the level of strength of the *cryptographic erase* operation.

Users seeking to leverage *cryptographic erase* should identify the following mechanisms the *storage device* implements to address these areas before relying on *cryptographic erase* for *media sanitization*:

— **Make/Model/Version/Media Type:** The product and versions the statement applies to, and the type of *storage media* the *storage device* uses (i.e., magnetic, SSD, hybrid, and other);

- **Key Generation:** Identify whether a deterministic random bit generator (e.g., one listed in NIST SP800-90A Revision 1 [B9]) was used, and how it has been validated;
- **Media Encryption:** Identify the algorithm, key strength, mode of operation, and any applicable validation(s);
- **Key Wrapping:** Identify whether the MEK (either wrapped with a KEK or not) is directly *sanitized*, or whether a key that wraps the MEK (a Key Encryption Key or KEK) is *sanitized*. A description of the wrapping techniques only applies where a KEK (and not the MEK) is *sanitized*. Wrapping details, when provided, should include the algorithm used, strength, and (if applicable) mode of operation;
- **Media Areas Addressed:** Describe which areas are encrypted and which areas are not encrypted. For any unencrypted areas, describe how *sanitization* is performed;
- **Key Life Cycle Management:** The key(s) on a *storage device* can have multiple wrapping activities (wrapping, unwrapping, and rewrapping) throughout the *storage device*'s lifecycle. Identify how the key(s) being *sanitized* are handled during wrapping activities not directly part of the *cryptographic erase* operation. For example, a user can have received an SED that was always encrypting and can have simply turned on the authentication function. Identify how the previous instance of the MEK was *sanitized* when it was wrapped with the user's authentication credentials;
- **Key Sanitization Technique:** Describe the *storage-media*-dependent *sanitization* method for the key being *sanitized*. Some examples might include three inverted overwrite passes if the *storage media* is magnetic, a block erase for an SSD, or other *media*-specific techniques for other types of *storage media*;
- **Key Escrow or Injection:** Identify whether the *storage device* supports key escrow or injection at or below the level of *cryptographic erase*. Identify whether the *storage device* supports discovery of whether any key(s) at or below the level of the key escrowed has/have ever been escrowed from or injected into the *storage device*. If the MEK encryption key is directly *sanitized* and only a KEK can be escrowed, clearly identify that fact;
- **Error Condition Handling:** Identify how the *storage device* handles error conditions that prevent the *cryptographic erase* operation from fully completing, such as if a defect is encountered where an instance of the key to be *sanitized* is stored. For example, if the location where the key was stored cannot be *sanitized*, the *cryptographic erase* operation can report success or failure to the user; and
- **Interface Clarity:** Identify which *host interface* commands support the features described in the statement. If the *storage device* supports the use of multiple MEKs, identify whether all MEKs are changed using the *host interface* commands available and any additional commands or actions necessary to ensure all MEKs are changed.

For all *storage device*s supporting encryption where *cryptographic erase* is intended for use to *purge* the *storage media* (including SEDs, mobile *storage device*s, and other *storage device*s), the level of assurance depends (in large part) on the following:

- the level of entropy of the MEK;
- if the key *sanitized* during *cryptographic erase* is a key that wraps the MEK (and not the MEK itself), the strength of the wrapping mechanism(s) and entropy of the wrapping key(s) to be *sanitized*;
- the strength of the encryption algorithm used to encrypt the data, including mode of operation and assurance of correct implementation; and
- the level of difficulty in retrieving the MEK after *sanitization*, plus any effort to unwrap the key (if it was stored wrapped with another value).

Mobile *storage devices* (and *storage device*s other than SEDs) can also support strong encryption capabilities. The decision regarding whether to rely on *cryptographic erase* to *purge* the *storage media* on those *storage device*s depends, in part, on whether all sensitive data are encrypted on the *storage device*. If encryption was enabled after sensitive data were stored on the *storage device*, or if it is unknown whether sensitive data were stored on the *storage device* prior to encryption, *cryptographic erase* is not appropriate as a *purge* mechanism.

An important issue in cryptographic erasure is destroying all copies of the encryption keys. SEDs do this by generating keys internally and never exposing them. Erasure is performed via a key change request operation. It is possible that SEDs contain multiple partitions, each with a unique key, thus requiring multiple key change requests.

Annex C

(informative)

Developing storage technologies

The field of storage technology is always creating new types of *storage media* and *storage device*s that contain *storage media*. This standard does not make any recommendations for security for *storage* technologies not in this standard. Some examples of developing storage technologies are as follows:

- persistent memory (e.g., NVDIMM-N);
- energy assisted magnetic recording (e.g., HAMR and MAMR);
- DNA storage;
- logical storage (e.g., cloud storage);
- holographic storage;
- storage attached to a fabric (e.g., SAN);
- object storage (e.g., key-value);
- encrypted storage with keys managed outside of the *storage device*;
- quantum cryptography;
- medical equipment; and
- automotive equipment.

Annex D

(informative)

Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[B1] INCITS 481-2011 (R2021), Information technology—Fibre Channel Protocol For SCSI-4 (FCP-4).[11]

[B2] INCITS 481-2011/AM1-2018, Information technology—Fibre Channel Protocol For SCSI, Fourth Version (FCP-4)—Amendment 1.

[B3] INCITS 502-2019, Information technology—SCSI Primary Commands—5 (SPC-5).

[B4] INCITS 506-2020, Information technology—SCSI Block Commands—4 (SBC-4).

[B5] INCITS 549-2021, Information technology—Zoned Device ATA Command Set—2 (ZAC-2).

[B6] INCITS 550, Information technology—Zoned Block Commands—2 (ZBC-2).

[B7] INCITS 558-2021, Information technology—ATA Command Set—5 (ACS-5).

[B8] ISO/IEC 17760-102:2016, Information Technology—AT Attachment—Part 102: ATA/ATAPI Command Set—2 (ACS-2).[12]

[B9] NIST SP 800-90A Rev. 1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators.[13]

[B10] NVM Express Base Specification, Revision 2.0b.[14]

[B11] NVM Express Key Value Command Set Specification 1.0b.

[B12] NVM Express NVM Command Set Specification, Revision 1.0b.

[B13] NVM Express Zoned Namespace Command Set Specification, Revision 1.1b.

[B14] NVM Express Management Interface Revision 1.2b.

[B15] Serial ATA Revision 3.5a.[15]

[B16] TCG Security Subsystem Class: Enterprise.[16]

[B17] TCG Security Subsystem Class: Opal.

[B18] TCG Security Subsystem Class: Opal 2.02.

[B19] TCG Security Subsystem Class: Opalite.

[B20] TCG Security Subsystem Class: Pyrite.

[B21] TCG Security Subsystem Class: Pyrite 2.0.

[11] INCITS publications are available from the InterNational Committee for Information Technology Standards (https://www.incits.org/).
[12] ISO publications are available from the International Organization for Standardization (https://www.iso.org/) and the American National Standards Institute (https://www.ansi.org/). IEC publications are available from the International Electrotechnical Commission (https://www.iec.ch) and the American National Standards Institute (https://www.ansi.org/).
[13] NIST publications are available from the National Institute of Standards and Technology (https://www.nist.gov/).
[14] NVM Express publications are available from NVM Express (https://nvmexpress.org/).
[15] Serial ATA publications are available from the Serial ATA International Organization (https://sata-io.org).
[16] TCG publications are available from the Trusted Computing Group (https://trustedcomputinggroup.org).

[B22] TCG Security Subsystem Class: Ruby.

[B23] TCG Storage Interface Interactions Specification.

IEEE SA
STANDARDS ASSOCIATION

RAISING THE WORLD'S STANDARDS

Connect with us on:

Twitter: twitter.com/ieeesa
Facebook: facebook.com/ieeesa
LinkedIn: linkedin.com/groups/1791118
Beyond Standards blog: beyondstandards.ieee.org
YouTube: youtube.com/ieeesa

standards.ieee.org
Phone: +1 732 981 0060

IEEE