Information Theoretic and Statistical Drive Sanitization Models

Jeffrey Medsger, Avinash Srinivasan, and Jie Wu

Abstract—Current enterprise drive sanitization techniques employ little or no intelligence to determine if the area being sanitized, with data overwriting, actually contains sensitive resident data. All data blocks in the target area are sanitized, utilizing brute-force sanitization techniques of one to several wipe passes. In reality, a significant number of drives needing sanitization may contain areas with no sensitive data, or even any data for that matter. Consequently, sanitizing such areas is counter-intuitive and counter-productive. In this paper, we propose two information-theoretic techniques – ERASE and ERASERS, which utilize an entropy measurement of data blocks for quick and effective drive sanitization. Our first technique, ERASE, unlike current brute-force methods, computes the entropy of each data block in the target area. Subsequently, all data blocks, which have an entropy within the user-specified sensitivity range, are wiped. Our second technique, ERASERS, which is an extension of ERASE, employs random sampling to enhance the speed performance of ERASE. To achieve this, ERASERS divides the target area into subpopulations, performs random sampling of blocks from each subpopulation, and computes the entropy of each sampled block. If the entropy of any sampled block, within a subpopulation, is within the user-specified sensitive entropy range, the entire subpopulation is wiped. The random sampling component of ERASERS gives organizations an alternative for a faster wipe, compared to the currently employed brute-force sanitization techniques. We have presented results, which compare the performance of our proposed techniques against the current brute-force technique. In a test, performed on the HFS+ unallocated space of an Apple MacBook Pro, used under real-world conditions, ERASERS averaged a speed improvement of 50.47% over a brute force technique, while retaining an accuracy of 99.84%, when set to a greater than 0 bpB and less than or equal to 8 bpB entropy range.

Index Terms—Drive Sanitization, Information Security, Information Theory, Random Sampling, Digital Forensics, Privacy, Entropy, Wiping.

1 Introduction

Deleted data remains on storage media until it is purged or overwritten completely. It is important to note that when a file is deleted, only its name is removed from the directory structure. Subsequently, the user data remains in the storage blocks of the drive, where it can be retrieved using forensic tools. Reformatting a hard disk drive only clears the file directory and severs the links between storage blocks. Such remnant data can eventually be recovered as long as it is not overwritten by new data, posing significant danger to users' security and privacy. Therefore, storage media sanitization is an important aspect in information security and digital forensics.

Today, security and privacy of consumer data is one of the biggest concerns for the computing industry. Legal requirements, mandating the protection of personal data from unauthorized access and disclosure, are far more stringent today than ever before. For instance, the enactment of laws, such as the *Health Insurance Portability and Accountability Act* (HIPPA) to protect the privacy of patients' health records creates extensive compliance concerns for healthcare information technology professionals. Under HIPPA, unauthorized disclosure of protected information can subject an organization to legal liability, negative publicity, monetary damages, and even criminal penalties. There are numerous laws and regulations that relate to data protection and privacy throughout the world, including, but not limited to – Data Protection Act of 1998 (UK), Financial Services Modernization Act (USA), and Personal Information Protection and Electronic Documents Act (Canada).

In light of the above discussions and from a user security and privacy perspective, it is extremely vital to protect the data at the end of its useful life cycle. Hence, data destruction is both essential and inevitable. However, we have seen that the amount of time required for sanitization is often long, and only increases with the ever increasing drive sizes, coupled with the ever decreasing costs.

1.1 Deleted Files

Studies indicate that simply deleting files and reformatting a hard drive is not adequate to erase all data such that it cannot be recovered [1]. In reality, data is rarely truly lost unless it is intentionally destroyed using one to several overwrites by special programs designed to remove all traces of the data. Data overwriting is a sanitization technique in which every bit of the target area is overwritten either using software or a hardware device. From a security

J. Medsger is an independent security and forensics researcher. email: medsgerj@gmail.com
A. Srinivasan is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA. e-mail: avinash@temple.edu
J. Wu is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA. email: jiewu@temple.edu

perspective, drive sanitization is vital when storage media leaves the control of its owner. Although encryption of data can provide a high degree of protection under some circumstances, it should not be relied upon in lieu of sanitization. Particularly, with the recent advances in password cracking and encryption brute-forcing using massively parallel GPU and cloud-based resources, sanitization is more critical than ever before for preserving the confidentiality of sensitive data on storage drives.

1.2 Types of Sanitization

Data destruction, a.k.a. *Sanitization*, can be primarily classified into two categories [2]:

1) *Destructive Sanitization:* provides user data confidentiality by physically destroying the storage device.
    - Physical Destruction: While "physical destruction" of storage devices offers the highest level of security, the number of used hard drives entering the secondary market, as noted in [2], is substantially large and dictates the use of non-destructive techniques. Physical destruction includes techniques, such as - breaking, chemical alteration, and phase transition of the media to be sanitized.
2) *Non-destructive Sanitization:* In this method, the confidentiality of user data is protected without destroying the storage media, such that the media can be reused.
    - Degaussing: It is the process of decreasing or eliminating a remnant magnetic field. Consequently, this destroys the data held on magnetic data storage.
    - Data Overwriting/Erasing: It is the process of overwriting every bit on the drive one to several times. The bit pattern used in overwriting can be all zeros, all ones, pseudorandom patterns, etc. This capability is built into existing command line utilities, such as *dd* and *dcfldd*.

The problem of data remanence is also countered by encryption, a technique referred to as crypto erase. Though this is not a sanitization technique, it does provide confidentiality of sensitive information stored on the storage device. Simply encrypting data before it is stored on the medium can mitigate inadvertent disclosure, if not eliminate it. Particularly, if the decryption key is strong and carefully controlled, it may effectively make any data on the medium unrecoverable. Even if the encryption key itself is on the medium, it is simpler to overwrite just the key rather than the entire disk – a technique currently embraced by the mobile device manufacturers. However, as previously mentioned, with current computing resources available, encryption will eventually be brute-forced if and when the drive leaves the control of its owner. Hence, encryption is not a feasible alternative for sanitization.

1.3 Adversarial Model

We assume that the adversary is technically sophisticated and has the necessary skills and tools to retrieve data from formatted drives, including from deleted partitions. It is also reasonable to assume that the adversary has knowledge of the user/system they are attacking, and can make use of this knowledge to perform string searches. The focus of our research, presented in this paper, is data overwriting – a non-destructive sanitization technique for magnetic storage media, specifically computer hard drives. Non-destrutive sanitization techniques, to their best abilities, enable reuse of the storage media, while seeking to prevent an adversary from recovering data from a used hard drive. Under this model, data privacy and security is assured irrespective of the type of access – logical, raw disk, or physical – the adversary has to the computer media. The adversary can make use of any techniques or tools they may have, post sanitization, to retrieve the data from the sanitized blocks, ranging from simple keyboard-based techniques (e.g. [2]) to sophisticated laboratory-based (e.g. [3], [4], [5]) methods/attacks to recover data.

1.4 Drive Sanitization and Digital Forensics

Drive sanitization is also an important process for digital forensics investigation, specifically for evidence acquisition. A previously-used drive has to be thoroughly sanitized, prior to acquiring evidence and/or starting a new case. This is very critical in preventing cross-contamination of evidence. The evidence could potentially be inadmissible in legal proceedings. Evidence, supporting sanitization of the acquisition disk, can and will be asked by the defense counsel, and has to be provided. Failure to do so, will render the evidence, and possibly the entire case, inadmissible. Sanitization evidence can be provided using cryptographic hash values, which can be verified. Most crime labs, that conduct digital forensics investigations, have a limited number of specialized machines, usually made-to-order, with all the necessary forensics tools, including the very expensive commercial tools. Consequently, they reuse a handful of hard drives on the same machine. This further justifies our attempt to propose more efficient and time-sensitive drive sanitization techniques.

On a separate note, sanitization can also be an indication of foul play by the adversary. It could be an antiforensics[1] attack by the adversary to eliminate their footprints, destroy data, or other such malicious intention. This, by far, is the most detrimental to a forensics investigation, wherein all evidentiary data is lost irreversibly. A forensics investigator is often trained, and has an eye for identifying string patterns – all zeros, random patterns with high entropy on an entire drive/partition, all ones, etc. – when blocks are sanitized. This can save precious time, that would otherwise be spent trying to analyze a sanitized disk with little evidence for prosecution. Note that an entire drive with random data, identified by a high entropy value, can also be

1. Antiforensics is an attack that is aimed at making the investigation extremely hard and frustrating to the investigator.

an indication of encrypted drives – BitLocker, TrueCrypt, etc.

Finally, within digital forensics, there is zero tolerance for data remanence on acquisition drives to prevent evidence contamination. Hence, for forensics applications, we can revert to current brute force techniques, a capability built into our ERASE tool.

1.5 Contributions

This paper is an extended version of [6]. This work was motivated by a need to optimize the drive sanitization process, specifically the data overwriting method of non-destructive drive sanitization. The contributions of this work can be summarized as follows:

1) To the best of our knowledge, this work is the first to do the following:
    a) Propose an entropy-based wiping method, for sanitizing a target area. The proposed method is a non-destructive drive sanitization method.
    b) Include random sampling, in tandem with the above entropy-based technique, for further enhancing the efficiency of sanitizing the given target area.
    c) Implement command line tools, with the aforementioned entropy measurement and random sampling capabilities.
    d) Evaluate the accuracy of the proposed ERASERS technique on a hard drive, used under real world conditions prior to the experiments.
    e) Evaluate the unallocated space of three hard drives, to analyze the distribution of the entropy of blocks over unallocated space.
    f) Analyze the common file types encountered on three real world hard drives, and compute the average file entropy of the top-50 most frequently occurring file types.
    g) Empirically analyze the minimum possible entropy of any given 4 KB block, for all possible words in a dictionary (UNIX dictionary). This result is also helpful in building the entropy baseline for other empirical studies.

1.6 Road Map

The rest of this paper is organized as follows. We begin with a discussion of the current state of affairs in drive sanitization in section 2. Then, in section 3, we present a brief discussion on the utility of the proposed information theoretic sanitization techniques – ERASE and ERASERS. We then provide brief discussions on relevant background areas in section 4, including random sampling, Shannon's entropy, and sanitization, for the purpose of completeness of this paper. In section 5, we provide details of the proposed sanitization techniques, ERASE and ERASERS, including wipe patterns, number of passes, and sampling parameter choices. In this section, we also discuss the impact of remnant data in the unallocated space on user-privacy and security, and the importance of its sanitization. Section 5 is followed by performance evaluation details and results, from tests under controlled conditions, in section 6. In section 7, we discuss the theoretical thoroughness of the ERASERS technique. In section 8, we discuss tests performed on three hard drives to better understand the distribution of data across unallocated space. In section 9, we discuss the performance and accuracy of ERASERS, when tested on the unallocated space of a hard drive, used under real world conditions, prior to the experiments. We then discuss related works in section 10 and conclude the paper with directions for future research in section 11.

2 Drive Sanitization- State of Affairs

Current approaches to remove resident data on hard drives take a brute force approach of overwriting the target area, in its entirety, in one to several passes. With drive sizes increasing to multiple terabyte scale, this brute force approach is becoming near impossible to accomplish in a timely manner. According to Seagate, the average size of hard drives they shipped in the year 2011 was 590 GB [7]. Through empirical analysis, we have determined that overwriting a 590 GB hard drive – with a single pass of random data using the command line tool dd, with $/dev/urandom$ as the *random input* source – takes approximately 14.6 hours. Similarly, if a single pass wipe is performed using the command line tool dd, with $/dev/zero$ as the input source, it takes approximately 1.584 hours to sanitize a 590 GB hard drive. Both of these tests were run on a machine, with the specifications presented in Table 4.

Unfortunately, with the current wiping methods, little or no intelligence is used to determine if the area being sanitized, with data overwriting, actually contains sensitive resident data. Instead, all data blocks in the target area are blindly sanitized. In reality, a significant number of drives needing sanitization may contain areas with no sensitive data or even any data for that matter. Consequently, sanitizing such areas is counter-intuitive and counter-productive.

Therefore, we propose ERASE, which uses entropy calculations to identify sensitive data in target areas, and overwrites only specific areas which are found to have sensitive data. ERASE optimizes the data overwriting process, allowing a 590 GB hard drive to be sanitized with 1 pass of a random pattern (/dev/urandom) in 9.5 hours, assuming 50% of the data on the drive is within the sensitive entropy range, on a machine with similar specifications to the one in Table 4. Thereby, ERASE achieves an average case performance improvement of approximately 34.8%.

ERASE can be further enhanced to improve performance times by using random sampling to identify data blocks containing sensitive data, within the specified target area, that need overwriting. We call the enhanced version ERASERS. With empirical analysis, we have determined that ERASERS takes 0.85 hours in its best-case performance, and 1.580 hours in its worst-case performance to sanitize a 590 GB drive using an input source of

$/dev/zero$. In the study we have assumed that 50% of the data is within the sensitive entropy range, on a machine with specifications similar to that presented in Table 4.

3 Utility of The Proposed Technique

3.1 ERASE

ERASE is designed to allow for a more efficient wipe in situations where pseudorandom data is being used as the wipe pattern. A pseudorandom source, such as $/dev/urandom$, is slow to generate data, and becomes the bottleneck of the wipe process. ERASE can be more efficient when wiping with pseudorandom data because it is able to reduce the number of writes by performing extra reads, allowing it to make a decision which blocks to overwrite, and overwriting only those blocks which have sensitive data. Thus, ERASE can perform a read operation (106.8 MB/s) to possibly avoid a write operation using data from $/dev/urandom$ (11.8 MB/s). ERASE is also designed to allow for a more efficient wipe when multiple overwrites are to be performed. For multi-pass wipes, ERASE performs a read (106.8 MB/s) to possibly prevent, for example, 3 overwrites (each at 105.8 MB/ s), if that area does not contain sensitive data necessary to be wiped.

3.2 ERASERS

The limitation of ERASE is that if a uniform pattern is used as the wipe pattern, such as /dev/zero, ERASE will not be more efficient when only one overwrite pass is desired. This is because a uniform source is significantly faster to generate than a pseudorandom source, causing the write rate to reflect the write speed of the drive, and not the rate at which pseudorandom data is generated. This causes the write speed of the drive to be the bottleneck of the wipe process and not the generation of pseudorandom data. Consequently, if the read and write speeds of the drive are the same, the extra reads, which ERASE performs, causes it to be less efficient than a brute force wipe, regardless of the amount of the area to be sanitized, which is in the sensitive entropy range. Therefore, ERASERS was designed to use random sampling to reduce the number of read operations that need to be performed in determining if blocks have sensitive data. ERASERS does not have to read the entire area, but instead randomly samples from parts (subpopulations) of the area (population) to determine which subpopulations may contain sensitive data. Subsequently, only those subpopulations are wiped, avoiding wiping of the entire population. We envision that ERASERS could be used by organizations to periodically overwrite the unallocated space of employee computers.

4 Background

This section will discuss background topics to provide the reader with the fundamentals necessary to understand the overwriting technique described in this paper.

4.1 Sanitization Algorithms

Numerous algorithms are available for overwriting data on storage media, and several government organizations have established standards for overwriting, dictating the number of passes and writing patterns – ones, zeros, or random patterns. After overwriting, a hard drive is still physically functional and can accept formatting. According to Guttman [3], overwriting only prevents previous data from being read by the operating system; however, he contends that there are specialized equipment, such as magnetic microscopes, that can still manage to read the original data. There has been severe criticism of this idea, and there has been no practical evidence to support that this holds true with modern hard drives. However, we feel obligated to cite this paper and some other related works [4], [5], and [8] for completeness of our paper.

4.2 Shannon's Entropy

Shannon's *entropy* is a measure of uncertainty of a set of probabilities and it measures the extent of compressibility of the given data [9]. The representation of entropy as defined by Shannon [9], which is described in our paper, is represented in Equation 1. In Equation 1, i is a given symbol out of a possible 1 to n different symbols, and p is the probability of the occurrence of the i^{th} symbol. Given a set of symbols, the fewer the number of different symbols and the higher the occurrences of those symbols, the lower the entropy of that set of symbols will be.

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{1}$$

Suppose that a 4096-byte block on a hard drive consisted of a single type of symbol, say the digit zero, which repeats 4096 times to fill up the entire block. This would be a situation of the lowest possible entropy for the given block size of 4096 bytes. In contrast, suppose that a 4096-byte block on a hard drive consists of 256 different symbols, which is the maximum number of unique symbols that can possibly be represented in a byte. Now, suppose that each symbol occurrs 16 times to fill up the entire 4096-byte block. This would be the scenario of the maximum possible entropy, for the given block size of 4096 bytes.

4.3 Random Sampling

In ERASERS, random sampling is used to determine – within a certain $Confidence\ Level$ (CL) and $Confidence\ Interval$ (CI), which are supplied by the user – when n blocks are sampled, if no blocks are encountered that have an entropy within a defined entropy range. When that is true, the subpopulation sampled is not likely to contain sensitive data, and therefore, is determined not to be wiped. Equations 2 and 3 below are used for computing the sample size necessary to reach a desired CI and CL.

$$n_0 = \frac{Z^2 p(1-p)}{e^2} \tag{2}$$

$$n = \frac{n_0}{1 + \frac{(n_0 - 1)}{N}} \tag{3}$$

In Equation 2, n_0 is the sample size, Z is the standard score which represents the standard deviation from the population mean, e is the precision level derived from [(confidence interval)/100], and p is the estimated proportion and is set to the maximum variance of 0.5 [10]. In Equation 3, n is the sample size adjusted using finite population correction, n_0 is the unadjusted sample size obtained from Equation 2, and N is the population size [10].

5 ERASE - A Novel Wipe Technique

In this section, we discuss in detail the ERASE method for sanitizing a target area. ERASE uses entropy measurements to identify sensitive data blocks in the target area to be sanitized, and overwrites only those blocks that contain data within a specified sensitive entropy range. Before using ERASE, the following information must be specified by the user:

1) Number of wipe passes
2) Wipe pattern - e.g., $/dev/zero$ or $/dev/urandom$
3) Sensitive entropy range for the target area
4) Random sampling confidence level and interval (if using ERASERS)

5.1 Number of Passes: One vs Multiple

First, the user needs to determine the number of passes to overwrite the target area if sensitive data is found. Within the digital forensics and security community, there is considerable debate regarding the minimum number of passes necessary to completely rid magnetic media of traces of remnant data. According to NIST SP800-88 [11], *"studies have shown that most of todayś media can be effectively cleared by one overwrite."* NIST SP800-88 suggests the single pass wipe should be performed with random data, rather than zeroing the target area [11]. The original version of the DoD industrial security manual, DoD 5220.22-M recommends a 3-pass wipe, consisting of one pass of a uniform character, one pass of its complement, and one pass of random characters. The current DoD 5220.22-M does not specify the number of passes, or the pattern required. Numerous wipe pass recommendations exist, but a unified consensus of how many passes are necessary has yet to be established. Therefore, ERASE has the capability of providing the user with single or multi-pass wipes.

5.2 Wipe Pattern: Zeros vs. Random Patterns

The second step is for the user to specify the pattern for overwriting the areas found to contain sensitive data. There are two common patterns used in wiping hard drives – uniform sequences of characters and random sequences of characters. All zeros or another uniform character in the target area generally indicates one of two things: either that portion of the disk has not been used, or someone has sanitized or attempted to sanitize the disk. On the other hand, wiping with random characters makes it difficult to determine if the disk contains data with high entropy values, such as docx, pptx, zip, or pdf file types, as shown in Figure 1(b), or if the target area was wiped with one to several passes using random data patterns. Therefore, ERASE provides both types of patterns to the user as an option. In our empirical studies, we have used two sources of data for wiping, both available on UNIX systems– $/dev/zero$ and $/dev/urandom$.

5.3 Entropy Range and Data Sensitivity

Third, a sensitive entropy range is constructed. The sensitive entropy range or set of sensitive entropy ranges defines the range of entropy, which the user believes could hold sensitive data.

The sensitive entropy range could be set to encapsulate:

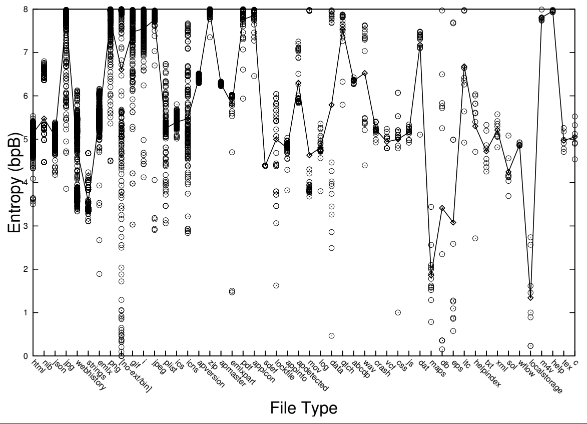
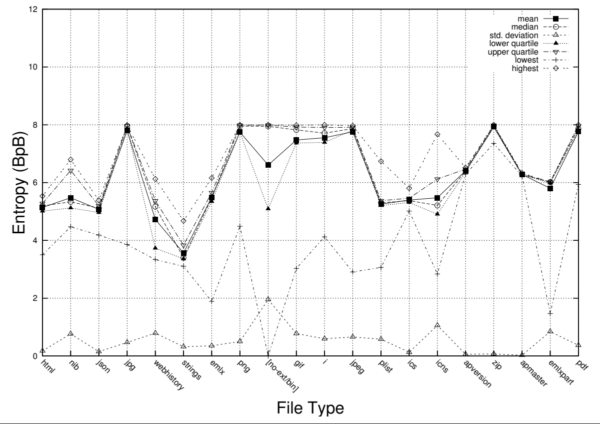
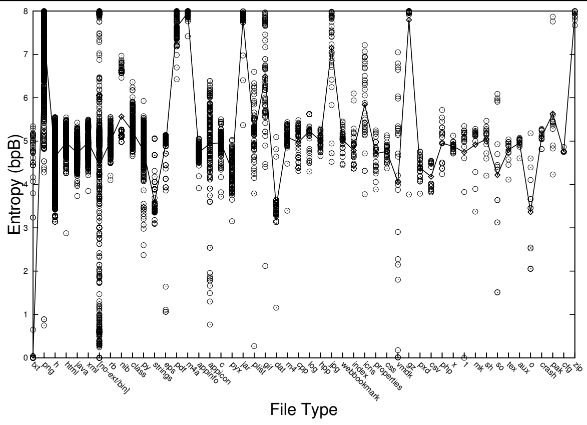
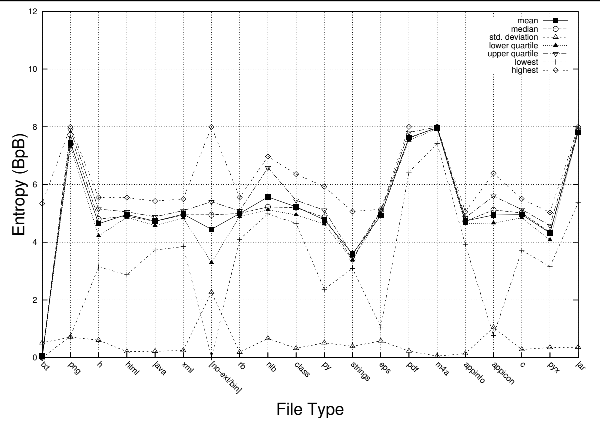
1) All blocks that have an entropy greater than zero. This would cause any block that consists of all uniform zeros or ones to not be wiped, and any block that is not all uniform ones or zeros to be wiped.
2) All blocks of a certain file type. Figure 1(b) displays the entropy of 30 common file types on a 6 GB partition of a Windows 7 machine in use for 2 years at the time of these experiments. Each data point in Figure 1(b) represents the entropy of an individual file on the system. The block size used to calculate the entropy of each file was equal to the size of the particular file being processed. Table 2 shows the entropy of randomly selected files under a user's home directory on two Macintosh OS X computers. Figures 2(a) and 2(f) show the block level entropy for six common file types.
3) All blocks that have an entropy between a very low entropy and a very high entropy. For example, a sensitive entropy range set to encapsulate all blocks that have an entropy equal to or higher than the lowest possible entropy that a block containing a certain length English word could contain. Fig. 1(a) shows, for each word in the Unix dictionary file, the lowest possible entropy that a 4096 B block could contain, if that block contained that word. Resultantly, if that word exists in a block on the drive, that block must have at least an entropy of that value.

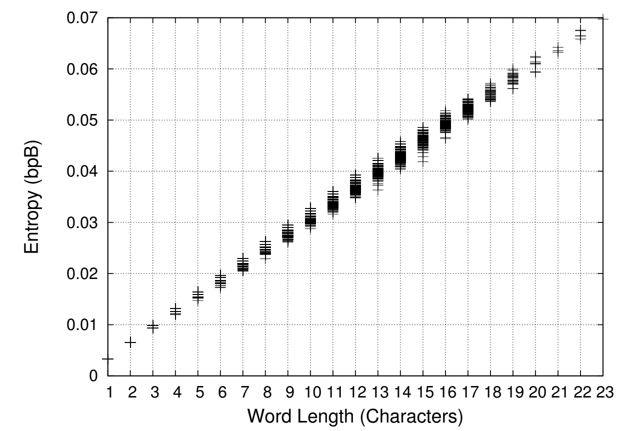
Additionally, by defining sensitive entropy ranges, blocks that fall within different sensitive entropy ranges, could be

TABLE 1
Specifications of the four machines used in the evaluations (shown in Tables 2, 5, and Figure 4)

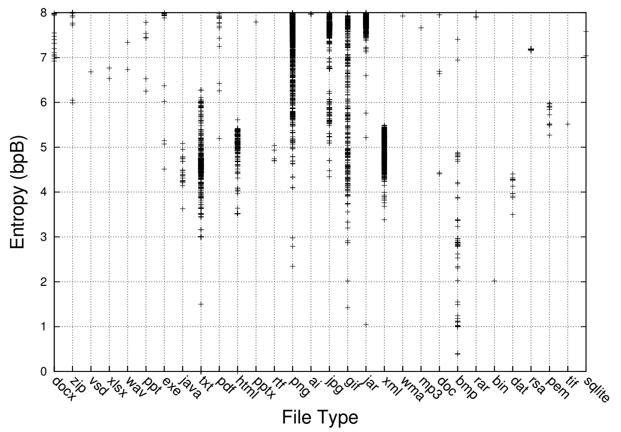
Machine	1	2	3	4
Type	iMac	Macbook Pro	Macbook Pro	Macbook Pro
OS	Mountain Lion	Snow Lepard	Lion	Mavericks
FS	HFS+	HFS+	HFS+	HFS+
Drive	HDD	HDD	SSD	HDD
Size	1 TB	300 GB	250 GB	300 GB
Use Life	6 months	30 months	14 months	3 months

TABLE 2
Entropy statistics of files encountered under home directories of two machines.

Machine	Entropy of randomly sampled files under one user's home directory (\Users\{username}) of a machine. Sampling performed with a 99% confidence level and 1% confidence interval. Results are shown for the top 50 most encountered file types. Averages for each file type are depicted by the plotted line.	Entropy statistics (mean, median, standard deviation, upper quartile, lower quartile, lowest, highest) computed for the top-20 most encountered file types during the test. Data points used in calculating the statistics were obtained from the test results shown in the graph to the left.
Machine 3, Reference Table 1		
Machine 2, Reference Table 1		

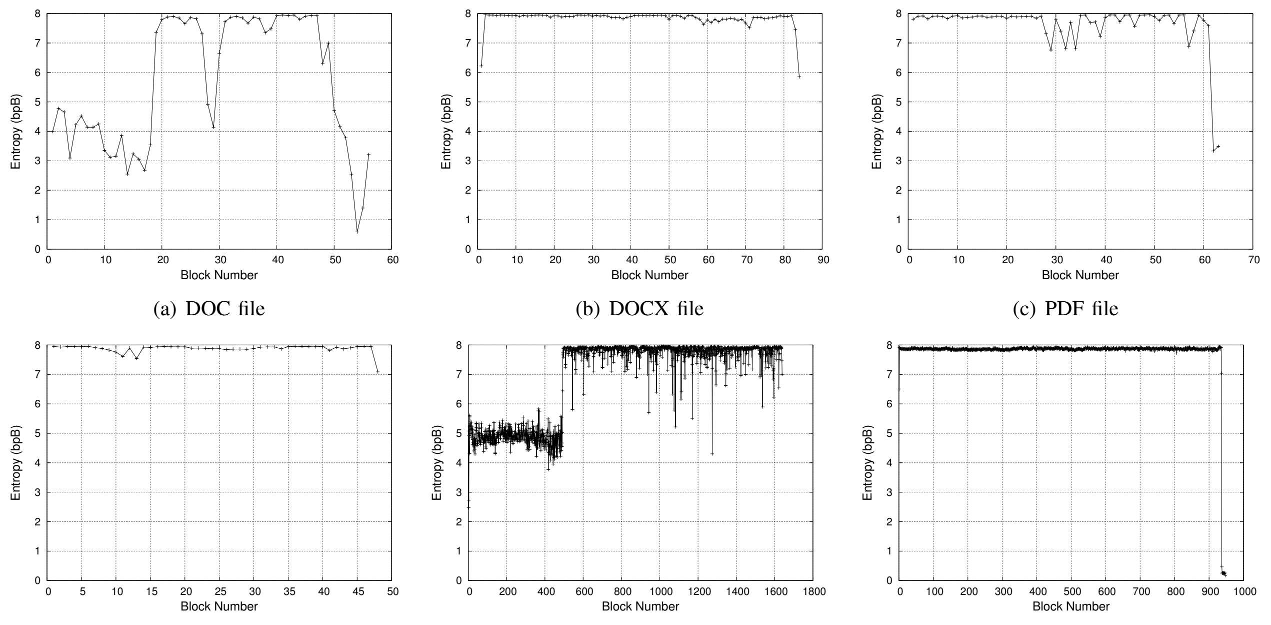


(a) Entropy of Dictionary Words



(b) Common File Type Entropy

Fig. 1. (a) Entropy of each word in a dictionary file when placed in a zeroized 4096-byte buffer, (b) Entropy of 30 most occurring file types found on a Windows-7 machine in use for 2 years.



(a) DOC file (b) DOCX file (c) PDF file

(d) ZIP file (e) SQLITE file (f) MP3 file

Fig. 2. Entropy values of all blocks of a given file computed for six commonly encountered file types.

TABLE 3
Variables used in this paper.

Equation 1 Variables	Equation 2 Variables	Equation 3 Variables
i = a given symbol	n_0 = sample size	n = sample size adjusted
n = total number of symbols	Z = standard score	(finite population correction)
p = probability of occurrence	e = precision level	n_0 = unadjusted sample size
	p = estimated proportion	N = population size
Equation 4 Variables	**Algorithm V.1: ERASE Algorithm**	**Algorithm V.2: ERASERS Algorithm**
p = probability	bN = block number	bL = block list
n = total number of blocks	tB = total number of blocks in area	bN = block number; [ ] = empty list
s = number of sensitive blocks	buf = disk block read into memory	tB = total number of blocks in area
z = sample size	e = entropy	sS = subpopulation size
	LEB = low sensitive entropy bound	sZ = sample size; r = random number
	HEB = high sensitive entropy bound	cL = confidence level; cI = confidence interval
		buf = disk block read into memory
		LEB = low sensitive entropy bound
		HEB = high sensitive entropy bound
		e = entropy; $\%$ = modulus operator
		rSS = regular subpopulation size

wiped with a different number of passes or wipe pattern, based on how sensitive users believe a block of data within that entropy range would normally be. For example, if a user feels that jpeg images, which are encoded causing them to have a high entropy, are more sensitive than a text document, which has a mid-range entropy, then higher entropy data could be overwritten with a more secure wipe pattern. The lower and upper entropy limits are tunable parameters in both ERASE and ERASERS and users can choose both the upper and lower bound values, according to their requirements.

After the sensitive entropy range is set, the wipe can be performed using either wipe methods presented in this paper. Each of the techniques are discussed in the following sections.

5.3.1 ERASE - Entropy Wipe No Sampling [EWNS]

In the ERASE (Entropy Wipe No Sampling [EWNS]) method, each block in the target area is read into a buffer, and each block's entropy is calculated. If a block has an entropy within the sensitive entropy range, then the block is wiped by overwriting that block p times, where p is the number of wipe passes.

5.3.2 ERASERS- Entropy Wipe Sampling [EWS]

The ERASERS (Entropy Wipe Sampling [EWS]) method uses both entropy calculations and random sampling. With ERASERS, the area to be sanitized is divided into a set of subpopulations. For each subpopulation, a sample size is determined based on the size of the subpopulation and the CL and CI the user specified. The sample size, n, indicates the number of blocks necessary to be sampled to meet the

CL and CI specified. Now, n blocks are randomly sampled from the subpopulation. Each sampled block is read into a buffer and its entropy is calculated. If any of the blocks sampled from a subpopulation contain data in the sensitive entropy range, the entire subpopulation is overwritten p times. This process is repeated for each subpopulation in the target area.

Algorithm 5.1: ERASE ALGORITHM($void$)

```
for $bN \leftarrow 0$ to $tB$
  do
    $buf \leftarrow$ READBLOCK($bN$)
    $e \leftarrow$ ENTROPY($buf$)
    if ($e >= LEB$ and $e <= HEB$)
     or ($e >= LEB2$ and $e <= HEB2$)
     or ...
      then OVERWRITEBLOCK($bN$)
```

Algorithm 5.2: ERASERS ALGORITHM($void$)

```
procedure GETSUBPOP($bN, tB, rSS$)
  $bL \leftarrow [\,]$
  for $i \leftarrow bN$ to $(bN + rSS)$
    do
      if $i < tB$
        then $bL.Add(i)$
        else return ($bL$)
  return ($bL$)

procedure OVERWRITESUBPOP($bL$)
  $sS \leftarrow$ LENGTH($bL$)
  for $i \leftarrow 0$ to $sS$
    do OVERWRITEBLOCK($bL[i]$)

main
  $bN \leftarrow 0$
  while LEN($bL \leftarrow$ GETSUBPOP($bN, tB, rSS$)) $> 0$
    do
      $sS \leftarrow$ LEN($bL$)
      $sZ \leftarrow$ SAMPLESIZE($sS, cL, cI$)
      for $i \leftarrow 0$ to $sZ$
        do
          $r \leftarrow$ GETRNDNUM() % $sS$
          $buf \leftarrow$ READBLOCK($bL[r]$)
          $e \leftarrow$ ENTROPY($buf$)
          if ($e >= LEB$ and $e <= HEB$)
           or ($e >= LEB2$ and $e <= HEB2$)
           or ...
            then OVERWRITESUBPOP($bL$)
      $bN \leftarrow bN + sS$
```

6 Performance Evaluation of Tests under Controlled Conditions

In this section, we discuss the performance of ERASE and ERASERS, when tested under controlled conditions, and compare them with the traditional brute force method.

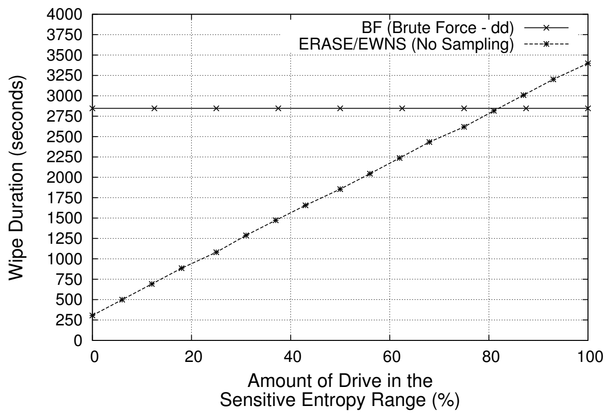
TABLE 4
Parameters of the machine used in the performance tests (shown in Figures 3(a) and 3(b))

Parameters of Machine	
Sequential Write ($/dev/zero$)	105.8 MB/s
Sequential Write ($/dev/urandom$)	11.8 MB/s
Sequential Read	106.8 MB/s
Avg. Random Seek Time	9.46 ms
Entropy Calc. Speed	5748.77 MB/s

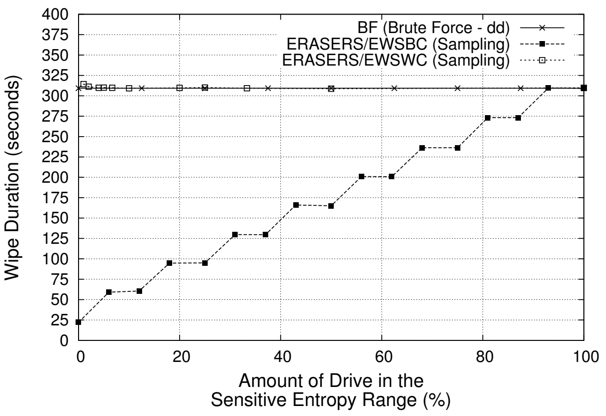
6.1 Performance Tests under Controlled Conditions

We performed the following tests on a desktop computer running Ubuntu Linux with an Intel(R) Core(TM) 2 Quad CPU 8300 @2.50GHz, 8 GB of RAM, and a Seagate ATA 2 TB 7200rpm hard drive. Table 4 summarizes the system's baseline read and write speeds, seek time, and entropy calculation speed. The partition used for the tests was 32 GB in size.

1) Brute Force (BF)– In the BF test, the entire partition was overwritten using dd, a Unix command line tool, with a block size of 4096 bytes.
2) ERASE (EWNS)– In the EWNS test, $x\%$ of the blocks in the partition were filled with data within the sensitive entropy range, and the rest were initialized with data outside the sensitive entropy range. Then, the tool executed in ERASE/EWNS mode using a block size of 4096 bytes.
3) ERASERS Best Case (EWSBC)– In the EWSBC test, $x\%$ of the blocks in the partition were sequentially filled with data within the sensitive entropy range, and the rest were initialized with data outside the sensitive entropy range. Then, the tool executed in ERASERS/EWS mode using a block size of 4096 B. A confidence level of 95% and a confidence interval of 5% were used in the random sampling by the tool. The number of subpopulations was set to $z = 8$ with each subpopulation being 4 GB in size. The EWSBC tests the best-case scenario for the ERASERS/EWS mode because random sampling is performed; however, only $\lceil((x\%/100) * z)\rceil$ subpopulations are overwritten, due to the sensitive data being clustered compactly and sequentially. For example, in the best-case test, if 50% of the drive is within the sensitive entropy range, then half of the subpopulations contain sensitive data and half of the subpopulations do not. Therefore, only half of the subpopulations need to be wiped.
4) ERASERS Worst Case (EWSWC)– In the EWSWC test, $x\%$ of the blocks in the partition were filled with data; however, as opposed to being sequentially filled as they were in the EWSBC test, they were spread out equally to fill the drive. For example, for the EWSWC test with 50% of blocks in the sensitive entropy range, every other block was filled with data in the sensitive entropy range; as opposed to the EWSBC test for



(a)



(b)

Fig. 3. (a) Comparison in performance between the Unix dd tool and the ERASE/Entropy Wipe No Sampling (EWNS) method. The wipe pattern used was 1 pass using $/dev/urandom$ as the source for each test using a 32 GB partition, (b) Comparison in performance between the Unix dd tool and the best and worst case performance of the ERASERS/Entropy Wipe Sampling (EWS) method. The wipe pattern used was 1 pass using $/dev/zero$ as the source for each test using a 32 GB partition. For the sampling method (EWS), the drive was divided into 8 sequential subpopulations of 4 GB each. A 95% confidence level and a 5% confidence interval were used in the sampling.

50%, which filled the first half of the drive with data in the sensitive entropy range, then the second half with data outside the sensitive entropy range. The EWSWC tests the worst-case scenario for the ERASERS/EWS mode because random sampling is performed; however, z subpopulations are overwritten because each subpopulation has some sensitive data in it.

We would like to draw the attention of the reader to the fact that entropy computation has a very minimum impact on performance, as can be seen from Table 4. Hence, using entropy computation for determining areas of sensitive data during sanitization is a justifiable approach, as it adds minimal overhead.

6.2 Performance Analysis of Tests under Controlled Conditions

Fig. 3(a) shows the performance of the ERASE/EWNS method compared to the performance of Unix tool, dd, when performing 1 pass using $/dev/urandom$ as the source. Fig. 3(b) shows the performance of the ERASERS/EWS method in its best and worst case scenarios compared to dd, when performing 1 pass using $/dev/zero$ as the source. The performance of the ERASE/EWNS method is more efficient than dd when 82% or less of the target area to be sanitized is within the sensitive entropy range. The performance of the ERASERS/EWS method is more efficient in its best-case scenario when 92% or less of the target area to be sanitized is within the sensitive entropy range. In its worst-case scenario, ERASERS/EWS is similar to the performance of dd.

Some key observations of the performance results shown in Figures 3(a) and 3(b) are as follows:

1) With no sensitive blocks in the specified entropy range, EWSBC ($/dev/zero$) is $14x$ faster compared to the brute force technique ($/dev/zero$), while EWNS ($/dev/urandom$) is approximately $9.5x$ faster than the brute force technique ($/dev/urandom$).
2) With 30% of the target drive containing data in the specified sensitive entropy range, EWSBC (/dev/zero) is approximately $2.3x$ faster than the brute force technique ($/dev/zero$) and EWNS ($/dev/urandom$) is approximately $2.2x$ faster than the brute force technique ($/dev/urandom$).
3) With 50% of the target drive containing data in the specified sensitive entropy range, EWSBC ($/dev/zero$) is approximately $1.9x$ faster than the brute force technique ($/dev/zero$), and EWNS ($/dev/urandom$) is approximately $1.6x$ faster than the brute force technique ($/dev/urandom$).
4) These results are independent of the actual entropy range chosen by the user.

Of the above three factors, the density of block distribution is perhaps the most critical factor. For example, if 10% of a drive's unallocated blocks are within the sensitive entropy range, but those blocks are evenly distributed across the unallocated space, ERASERS will have close to worst-case performance. In that instance, all the subpopulations will contain one or more sensitive blocks, thereby necessitating all subpopulations to be overwritten. Under this scenario, performance of ERASERS will be similar to current brute force techniques. In contrast, if 10% of a drive's unallocated blocks are within the sensitive entropy

TABLE 5
Distribution of entropy of data blocks in the unallocated space of three machines.

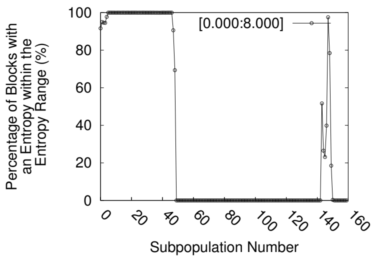
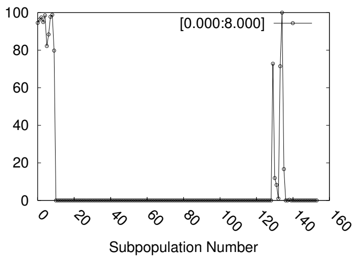
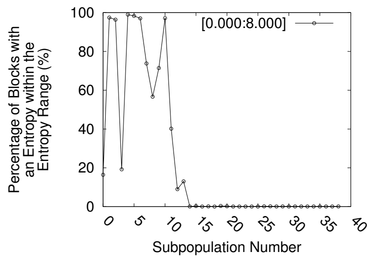
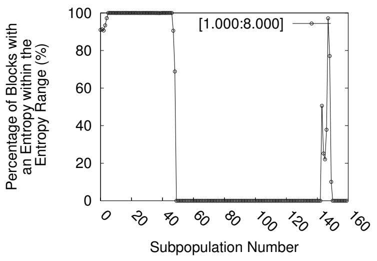
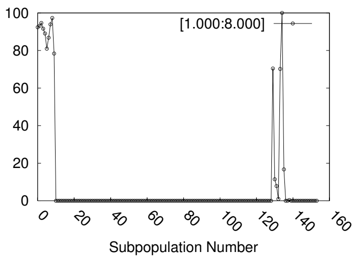
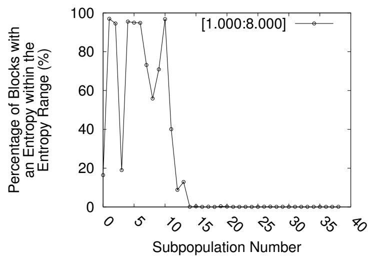
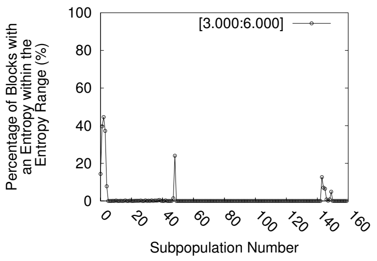
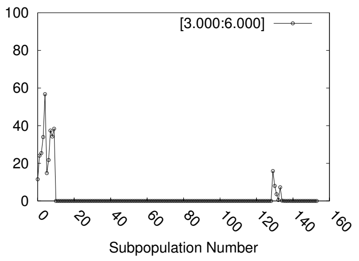
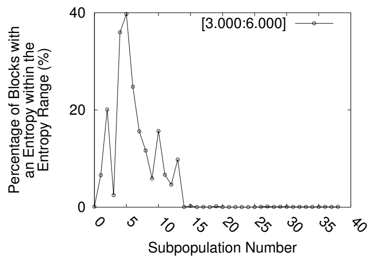
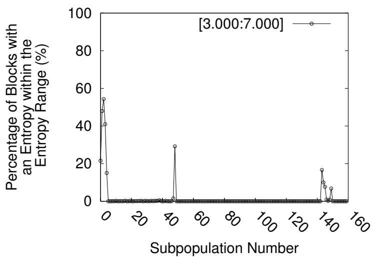
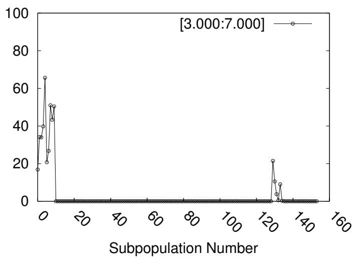
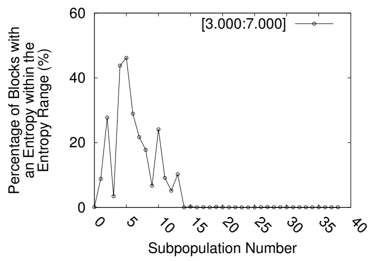
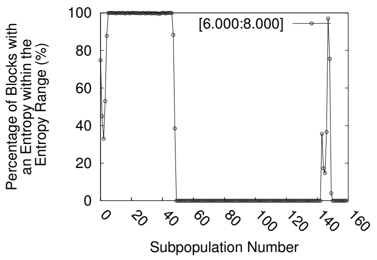
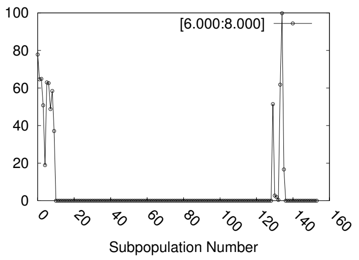
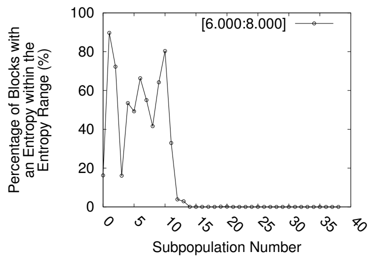
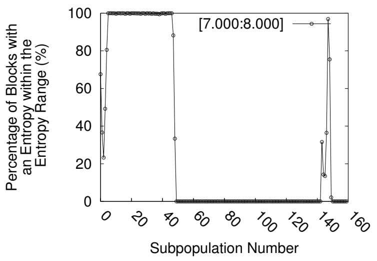
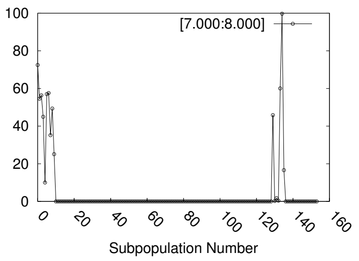
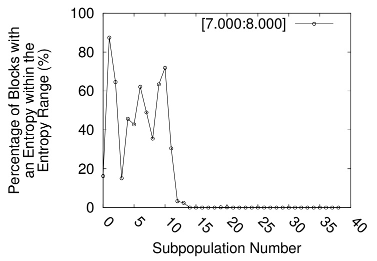
Entropy Range	Machine 1, Reference Table 1	Machine 3, Reference Table 1	Machine 4, Reference Table 1
$0-8$			
$1-8$			
$3-6$			
$3-7$			
$6-8$			
$7-8$			

TABLE 6
Thoroughness of technique evaluated using probability on a target area of $1,048,576$ blocks.

Number of sensitive (4096-byte) Blocks	10	100	1000	10000
Probability of not sampling a sensitive block	99.63%	96.40%	69.32%	2.52%
Percentage of population, that are sensitive blocks	0.00095%	0.0095%	0.095%	0.95%

range, but those sensitive blocks are tightly clustered (concentrated) in certain areas, then only a small number of subpopulations will have blocks within the specified sensitive entropy range. Consequently, ERASERS will have near best-case performance since only a small number of subpopulations will have to be overwritten. This is also more realistic because blocks of data, that belong to the same file, tend to have similar entropy values and often occur in continuity.

The performance of our proposed sanitization technique with random sampling – ERASERS – is strongly dependent on the following three factors:

1) number of data blocks in the target area
2) number of data blocks within the specified sensitive entropy range in the target area
3) density of distribution of sensitive data blocks within the target area

7 Theoretical Thoroughness of ERASERS

Table 6 shows the probability of not sampling (missing) a sensitive block, when 384 blocks are randomly sampled from a population of $1,048,576$ blocks. The results presented in Table 6 are for scenarios when there are 10, 100, 1000, and 10000 sensitive blocks in a population of 1048576 blocks and 384 blocks are randomly sampled. Here, we choose 384 sampled blocks and a population of $1,048,576$ blocks to keep evaluations meaningful and consistent with the settings used by the tool in the performance tests presented in section 6.2 and in Fig. 3(b). Equation 4 was used to calculate the probabilities.

$$p = 100.0 * \frac{\binom{n-s}{z}}{\binom{n}{z}} \tag{4}$$

8 Analysis of Unallocated Space

We conducted tests to understand the distribution of data in the unallocated space of three different HDDs, formatted with the HFS+ filesystem. In the tests, each drive's unallocated space was divided into 4 GB subpopulations, with each subpopulation consisting of sequential unallocated disk blocks. For example, Subpopulation-0 consisted of the first $1,048,576$ unallocated blocks ($blocksize = 4096B$), as sequentially encountered when reading the HFS+ allocation bitmap file. Subpopulation-2 consisted of the next $1,048,576$ unallocated blocks encountered in the allocation bitmap file, and the same repeated for the other subpopulations. The entropy of each block, in each subpopulation, was calculated. Subsequently, for each subpopulation, the percentage of blocks within a given entropy range was calculated for six different sensitive entropy ranges. The results, depicting the distribution, are shown in Table 5.

9 Evaluations under Real World Conditions

We evaluated the performance of the ERASERS tool on a Apple Macbook Pro with a 2.3 GHz Intel Core i5 processor, 4 GB of DDR3 RAM, and a 300 GB SATA hard disk drive. This computer is listed as Machine 4 in Table 1. A depiction of the distribution of the data in its unallocated space is shown in Table 5, Machine 4. An initial image of the main HFS+ partition of the hard drive was copied to an external 3TB hard drive with a Thunderbolt connection. The initial image was not modified in any of the tests. Before each test was performed, a second image was created of the first image, creating an exact bit-for-bit copy of the first image. Each test of the tool was performed on the second images. This ensured that each test was run on an image that had the exact same state as the initial image of the main partition. The main partition of the hard drive was 292.94 GB. 140.86 GB of the hard drive consisted of allocated space and 152.08 GB was unallocated. The goal of the tests was to evaluate the performance and accuracy of the ERASERS tool against a brute force tool (UBLKW) on the unallocated space of a HFS+ partition. The ERASERS tool was modified to parse HFS+ data structures to read the unallocated space of a HFS+ partition. A second tool was created, called UBLKW, which overwrites each unallocated block of a HFS+ partition with zeros, in a sequential order, according to the HFS+ allocation bitmap. The UBLKW represents the brute-force method of sanitization. The ERASERS tool was tested with a [0:8] bpB entropy range, which considered any block with an entropy greater than 0 bpB and less than or equal to 8 bpB as sensitive data. A [0:8] bpB entropy range is the most conservative setting of ERASERS, which considers all blocks, that do not contain completely uniform data (e.g. all zeros), as sensitive. The ERASERS tool was also tested with a [7:8] bpB entropy range. The results of the tests, in comparison to the UBLKW tool, are shown in Figures 4(a) and 4(b).

9.1 Performance and Accruacy

The ERASERS tool, when set to an entropy range of [0:8] bpB, averaged a 50.473% speed improvement over the brute-force UBLKW tool, over 12 iterations. The ERASERS tool averaged an accuracy of 99.845%, in the [0:8] bpB entropy range setting – meaning that 99.845% of the blocks, within the set entropy range, were sanitized.

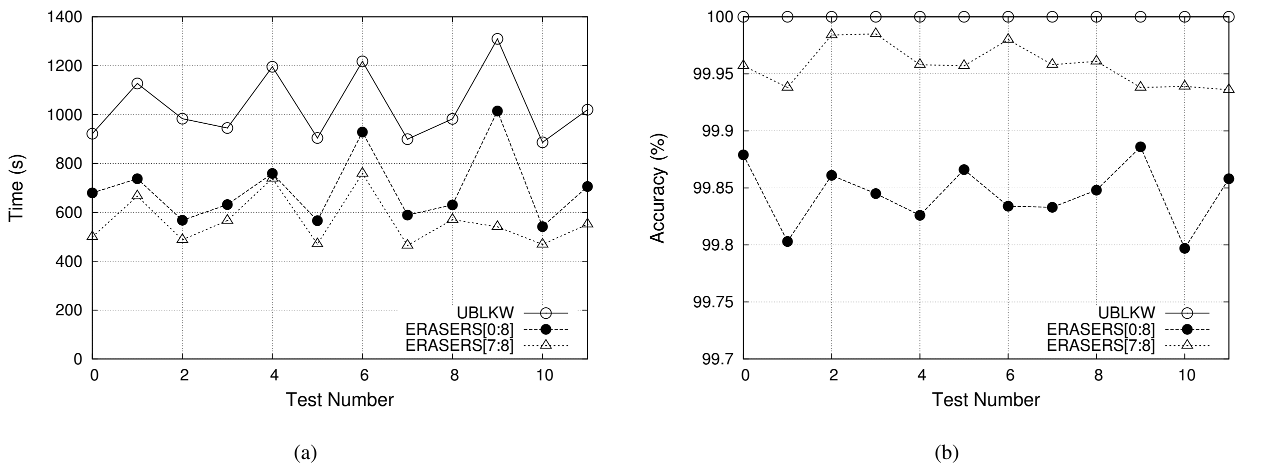


Fig. 4. (a) Performance of the UBLKW tool, ERASERS configured to an entropy range of greater than 0 bpB and less than or equal to 8 bpB, and ERASERS configured to an entropy range of greater than 7 bpB and less than or equal to 8 bpB. Uniform zeros were used as the wipe pattern in all of the tests. (b) Accuracy of the UBLKW tool, ERASERS configured to an entropy range of greater than 0 bpB and less than or equal to 8 bpB, and ERASERS configured to an entropy range of greater than 7 bpB and less than or equal to 8 bpB.

10 Related Work

The study in [2] indicates that the secondary hard-disk market is almost certainly awash in information that is both sensitive and confidential. With several months of work and relatively little financial expenditure, the authors of the study [2] were able to retrieve, from previously owned hard drives, thousands of credit card numbers and other personal information of many individuals. Garfinkel [2] believes few people are currently looking nefariously to secondary market hard drives for confidential data. However, Garfinkel recognizes that if sanitization practices are not significantly improved, it is only a matter of time before the confidential information on such hard drives will be targeted and exploited.

The authors in [12], note that when a computer is lost or disposed, active and discarded data typically remains stored on its hard disk drive. Even if users "delete" all their files, they can be easily recovered from "Recycling" and "Trash" folders, or by special utility programs, such as Norton Unerase or other forensics tools. The authors in [13] note that, for all but the most sensitive information, users will turn to efficient erasure methods that take minutes rather than hours or days. The selection of a wiping method will predominantly be driven by a tradeoff between an acceptable level of security risk and the time necessary to perform the overwriting. This is one of the strong motivating factors for the research presented in this paper.

The following related works present uses of random sampling to accelerate processes in computer security and forensics. Mora and Kloet, in [14], propose randomly sampling image (picture) files in a directory to allow a forensic analyst to identify a representative sample of the contents of the directory without having to manually review all images in the directory. Young et. al., in [15], propose randomly sampling sectors from computer media and calculating cryptographic hashes of the sampled sectors. The hashes of the sampled sectors are compared to a database, which contains sector-level hashes of known, illegal or malicious files. The method in [15] allows for a quick search to be performed using random sampling at the sector level to determine if a computer media contains files from a database of know, illegal files.

The following are related works in the area of computer media data sanitization. In [16], Reardon et. al. provide a survey of current secure data deletion methods. [17], [18], and [19] present works relating to secure deletion of individual files on computer media. In [17], Botelho et. al. provide a method for sanitizing deduplicated storage systems using perfect hashing. Joukov et. al., in [18], present a method of wiping individual files and their associated metadata on ext3 filesystems. In [19], Wei et. al. present a method of erasing individual files from flash-based solid state drives using flash translation layer extensions. In [20], Nisbet et. al. present an analysis of the effect of the TRIM command on the retention of residual data on solid state drives. In [21], Savoldi et. al. present statistical methods for detecting wiped areas on computer media. One method tested in [21] is the use of an entropy-based classifier to determine whether disk fragments were wiped with random data. In addition to the entropy classifier tested in [21], the authors in [21] implemented a classifier for detecting wiped areas on computer media using a statistical package from the National Institute of Standards and Technology (NIST). The NIST statistical package consists of 15 tests, which evaluate the randomness of data.

11 Conclusion

In this paper, we have presented two novel information theoretic drive sanitization techniques – ERASE and

ERASERS. The first proposed technique, ERASE, computes the entropy of blocks in the target area to be sanitized. Then, based on the computed entropy, a decision is made whether the blocks need to be sanitized. The second proposed technique, ERASERS, is an extension of ERASE, and uses random sampling to further enhance the performance of drive sanitization. It optimizes the ERASE wiping process using random sampling for a user-specified CL and CI.

As part of our future research, we intend to build an entropy baseline for data blocks of various file types encountered on mainstream desktop and server operating systems. This, we believe will help further enhance the performance of ERASE and ERASERS.

References

[1] M. Caltabiano, "Sanitization of computers and electronic storage media a.k.a. - disk sanitization," in *Office of Environmental Information, Office of Technology Operations and Planning*.

[2] S. Garfinkel and A. Shelat, "Remembrance of data passed: A study of disk sanitization practices," *IEEE Security and Privacy*, vol. 1, pp. 17–27, 2003.

[3] P. Gutmann, "Secure deletion of data from magnetic and solid-state memory," in *Proceedings of the 6th USENIX Security Symposium*, 1996, pp. 77–89.

[4] R. Gomez, A. Adly, I. Mayergoyz, and E. Burke, "Magnetic force scanning tunnelling microscope imaging of overwritten data." in *IEEE Transactions on Magnetics*, vol. 28, no. 5, 1992.

[5] R. Gomez, E. Burke, A. Adly, I. Mayergoyz, and J. Gorczyca, "Microscopic investigations of overwritten data," in *Journal of Applied Physics*, vol. 73, no. 10, 1993, pp. 6001–6003.

[6] J. Medsger and A. Srinivasan, "Erase- entropy-based sanitization of sensitive data for privacy preservation," in *Proceedings of The 7th International Conference for Internet Technology and Secured Transactions*, 2012, pp. 427–432.

[7] Seagate-Technology-LLC, "Seagate technology reports fiscal fourth quarter and year-end 2011 financial results," *About Seagate News Room*, Jul. 2011.

[8] C. Wright, D. Kleinman, and S. Sundhar, "Overwriting hard drive data: The great wiping controversy." in *Proceedings of the 4th International Conference on Information Systems Security*, 2008, pp. 244–257.

[9] C. E. Shannon, "A mathematical theory of communication," in *Bell System Technical Journal*, vol. 27, 1948, pp. 379–423 and 623–656.

[10] G. Israel, "Determining sample size." in *University of Florida IFAS Extension, PEOD6*, 1992.

[11] R. Kissel, M. Scholl, S. Skolochenko, and X. Li, "Guidelines for media sanitization: Recommendations of the national institute of standards and technology," in *NIST SP 800-88 Report*, 2006.

[12] G. Hughes, T. Coughlin, and D. Commins, "Disposal of disk and tape data by secure sanitization," *IEEE Security and Privacy*, vol. 7, no. 4, pp. 29–34, Jul. 2009. [Online]. Available: http://dx.doi.org/10.1109/MSP.2009.89

[13] G. Hughes and T. Coughlin, "Tutorial on disk drive data sanitization," in *UCSD CMRR Report*, 2007.

[14] R.-J. Mora and K. Bas, "Digital forensic sampling," *Sans Institute Publication*, 2010.

[15] J. Young, K. Foster, S. Garfinkel, and K. Fairbanks, "Distinct sector hashes for target file detection," *Computer*, 2012.

[16] J. Reardon, D. Basin, and S. Capkun, "Sok: Secure data deletion," in *SP '13 Proceedings of the 2013 IEEE Symposium on Security and Privacy*, 2013.

[17] F. C. Botelho, P. Shilane, N. Garg, and W. Hsu, "Memory efficient sanitization of a deduplicated storage system," in *Proceedings of the 11th USENIX Conference on File and Storage Technologies (FAST '13)*, 2013.

[18] N. Joukov, H. Papaxenopoulos, and E. Zadok, "Secure deletion myths, issues, and solutions," in *Proceedings of the ACM Workshop on Storage Security and Survivability*, 2006.

[19] M. Wei, L. Grupp, F. Spada, and S. Swanson, "Reliably erasing data from flash-based solid state drives," in *FAST'11 Proceedings of the 9th USENIX conference on File and stroage technologies*, 2011.

[20] A. Nisbet, S. Lawrence, and M. Ruff, "A forensic analysis and comparison of solid state drive data retention with trim enabled file systems," in *Proceedings of the Australian Digital Forensics Conferences*, 2013.

[21] A. Savoldi, M. Piccinelli, and P. Gubian, "A statistical method for detecting on-disk wiped areas," *Digital Investigation*, 2011.

Jeffrey Medsger received Bachelor of Science in Information Technology from George Mason University, Voglenau School of Engineering, in May 2011, and Master of Science in Information Security and Assurance from George Mason University, Voglenau School of Engineering, in May 2012. He has published two papers in the proceedings of international technology conferences, and received a best paper award at the 7th International Conference for Internet Technology and Secured Transactions (ICITST-2012).

Avinash Srinivasan is a faculty member in the Department of Computer and Information Sciences at Temple University. Dr. Srinivasan earned his B.E. (Industrial Production, 1999) from University of Mysore with Honors and M.S. (Computer Science, 2003) from Pace University, New York with *Distinguished Achievement Award for Academic Excellence*. He received Ph.D. in Computer Science from Florida Atlantic University in Aug. 2008, and Prof. Jie Wu (IEEE Fellow) was his advisor. He was a recipient of the two prestigious and competitive University-wide fellowships– *Dr. Daniel B. & Aural B. Newell Doctoral Fellowship* for the academic year 2005 - 2006 and *Graduate Fellowship for Academic Excellence* for the academic year 2006 - 2007. His research interests include network security and forensics, forensic analysis of file systems, forensic file carving, and security in WSNs & MANETs. He has published 34 refereed papers in scholarly conferences and journals, including IEEE INFOCOM, ACM SAC, IEEE MALWARE, IEEE ICDCS, and IEEE ICC. Dr. Srinivasan is a member of the *NIST Cloud Computing Forensics Science* and *NIST Cloud Computing Security* Working Groups.

Jie Wu is the chair and a Laura H. Carnell Professor in the Department of Computer and Information Sciences at Temple University. Prior to joining Temple University, he was a program director at the National Science Foundation and Distinguished Professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Computers, IEEE Transactions on Service Computing, and Journal of Parallel and Distributed Computing. Dr. Wu was general co-chair/chair for IEEE MASS 2006 and IEEE IPDPS 2008 and program co-chair for IEEE INFOCOM 2011. Currently, he is serving as general chair for IEEE ICDCS 2013 and ACM MobiHoc 2014, and program chair for CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.