Linnæus University
School of Computer Science, Physics and Mathematics

Examensarbete

Performance differences in encryption software versus storage devices

Author: Robin Olsson
Email: ro222ae@student.lnu.se
Supervisor: Martin Fredriksson
Semester: VT12
Course code: 1DV41E

Abstract

This thesis looked at three encryption applications that all use the symmetric encryption algorithms AES, Twofish and Serpent but differ in their implementation and how this difference would illustrate itself in performance benchmarks depending on the type of storage device that they were used on. Three mechanical hard drives and one solid state drive were used in the performance benchmarks which measured a variety of different disk operations across the three encryption applications and their algorithms. From the benchmarks performance charts were produced which showed that DiskCryptor had the best performance when using a solid state drive and that TrueCrypt had the best performance when using mechanical hard drives. By choosing DiskCryptor as the encryption application when using a solid state drive a performance increase of 38.9% compared to BestCrypt and 28.4% compared to TrueCrypt was achieve when using the AES algorithm. It was also shown that Twofish was overall the best performing algorithm. The primary conclusion that can be drawn from this thesis is that it is important to choose the right encryption application depending on the type of storage device used in order to get the best performance possible.

Keywords: Encryption, Symmetric Encryption, AES, Twofish, Serpent, TrueCrypt, BestCrypt, DiskCryptor, SSD, Mechanical hard drive, Performance, Benchmarks

Table of contents

1. Introduction

The widespread introduction of computers in companies all around the world saw an increased need to protect the assets of the company by using encryption as one of the main methods. As mobile devices such as laptops, notebooks, netbooks and smartphones become ever more popular in our everyday lives the risk of revealing sensitive information in the event of theft or loss makes encryption a viable option for everyone concerned about their privacy.

This thesis will evaluate three encryption applications that all use the same set of encryption algorithms but implement them in a different manner and how these implemental differences affect the performance on a variety of different storage devices [1].

1.1 Background

This thesis will cover which impact certain types of storage devices has on the implementation of encryption software available today and to be able to understand what encryption does there will be a slight summary of the goals and purpose of encryption given below.

The concept of encryption and the purpose of encryption is to transform a piece of information which is sometimes known as a plaintext into something that is unreadable if you do not possess the method to change the information back to its original state [1].

Ever since the fifth century B.C. when the Spartans invented the scytale, the first ever military cryptographic device, there has been a demand for the ability to securely transfer information over unsecure channels through the use of encryption [1]. In 1973 roughly 2500 years after the invention of the scytale the American government sought to find a method that would allow them to safely store sensitive government information. A solution to this problem was defined and endorsed by the American government in 1977 through the release of the encryption standard DES [2].

In 1997 it was shown that DES was vulnerable to brute force attacks which prompted a public request for the development of a new encryption standard by the National Institute of Standards and Technology [3, 4]. This request received a lot of attention as a total of fifteen algorithm designs from twelve different countries were submitted. From these fifteen only MARS, RC6, Rijndael, Serpent and Twofish were chosen as the finalists. Out of these five submissions Twofish came on third place, Serpent came on second place and Rijndael won. The winning algorithm was named the Advanced Encryption Standard (AES) [5].

As the scientific community finally received a new set of algorithms the question raised by myself is which of these applications that make use of these new algorithms present the best performance in regards to the storage device used.

1.2 Previous Research

Since little to no research has been put into examining the impact that the type of storage device used has in regards to the type of encryption applications and algorithms that is being used this chapter about previous research has only included research that is related to the area of encryption as a whole or to specific applications that will be evaluated during this thesis to show that the area is under research but has not yet ventured into the area that this thesis will cover.

Research made regarding TrueCrypt explains why it is important to use storage encryption and how it in theory is applied depending on if you’re using a USB-device or a hard drive and the different operational methods used in TrueCrypt such as the encryption of file systems, partitions and portable hard drives [6].

The second software that will be benchmarked in this paper is BestCrypt which was evaluated by Mick Bauer and published in the Linux Journal [7]. This paper is only an evaluation of BestCrypt and does not compare it to other available encryption applications.

The third research that is related to this thesis compared the six most commonly used symmetric algorithms which they claim to be AES, DES, 3DES, RC2, Blowfish and RC6 and measured the effect on CPU usage and battery life on laptops [8]. This is similar to what this thesis intends to do study except that it will focus on benchmarking AES, Twofish and Serpent and that it will focus on the performance impact of using different storage devices such as a solid state drives versus traditional mechanical hard drives as well as comparing the difference in performance depending on the encryption software and algorithm used.

The last research compared the performance of the DES, 3DES, AES and Blowfish algorithms by using two computers with different processors. This study is similar to this thesis in that it compares different symmetric encryption algorithms but it only focuses on the impact that the CPU has on the performance [9].

1.3 Problem Description

The main event that ultimately prompted a reason to study this area originated after having installed the encryption application TrueCrypt on a solid state drive which resulted in severe performance issues to such an extent that it would freeze the system for several seconds at a time and generate graphical glitches on the screen. This was a phenomenon that had not been encountered when using mechanical hard drives which sparked an intense urge to try and find the reason behind this particular problem and if it was something related to the solid state drive or the implementation of the application used. By referring to Figure 1 I believe that the problem lies between how the specific application decides to implement the algorithm in the software and how this particular piece of software is adapted to working on newer technology hardware such as a solid state drive and that they might not be updated to properly handle the different underlying architecture that the solid state drive uses. Since the algorithm is the same for all applications the only thing that can differ is the implementation and interaction with the hardware as illustrated in Figure 1.

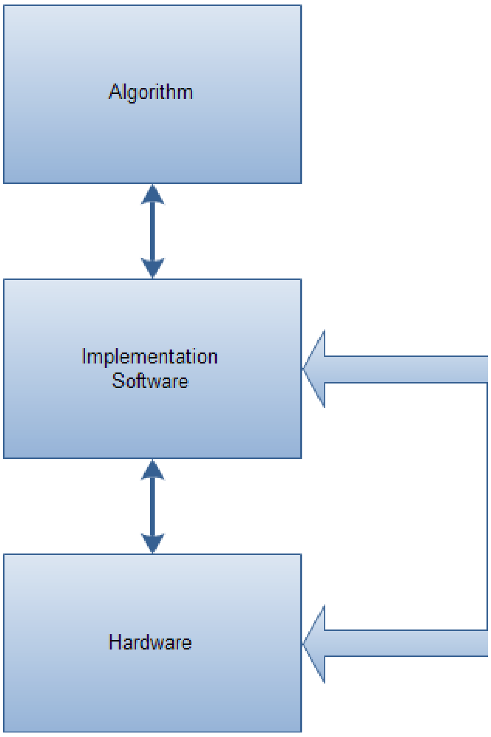


Figure 1: Logical Encryption Implementation Schematics

The secondary purpose of this thesis is to evaluate three encryption applications that all make use of AES, Twofish and the Serpent algorithm but implement them in different ways and how this difference affects the performance depending on the storage device used in particular the difference between using a normal mechanical hard drive and a solid state drive in order to find the best performing application and algorithm depending on the type of hardware you have.

This difference in implementation raises the question about which encryption application offers the best performance depending on the hardware used. As previous research has been focusing on the impact that the CPU has on performance this thesis will instead shift the focus to studying the performance of encryption depending on the specific storage device used which is something that has previously been neglected.

This thesis will also find out of there is any difference to how a solid state drive interacts with the different encryption applications and algorithms compared to its mechanical counterpart when it comes to performance and to see if the problems encountered when encrypting a solid state drive with TrueCrypt can illustrated in the performance benchmarks.

1.4 Limitations

The performance benchmarks in this thesis will cover the differences in performance depending on the storage device used but will not take into account the CPU which also has a big impact on the time it takes to encrypt and decrypt data on-the-fly especially when using a CPU with “AES-SI” that is a hardware accelerated encryption technology found in newer processors [10].

This study will look into a number of applications that can be used to encrypt a storage device but it will not cover all of them due to the sheer amount of available software that exist and the time limit set on this thesis, it will however give a good overview on a few of these applications.

The benchmarks in this thesis will only be performed on the Windows operating system because according to netmarketshare.com the Windows 7 operating system is being used by over 38% of users making it the second most widely used operating system and therefor has the widest audience.[1] Another reason for this limitation is the lack of support for operating systems such as Linux and Mac in the benchmark tools being used to perform the tests. The reason for not using the most popular operating system which is Windows XP is because it is in the process of being phased out in favor of Windows 7 and later Windows 8. The results in this thesis might still be applicable to Windows XP because of similarities in the underlying architecture of the operating system when comparing it to Windows 7 and that the encryption applications all support it [11].

It is important to note that the benchmarks will be performed on consumer rated electronics and not server rated electronics which means that the performance and reliability of the tested equipment should be lower than the high-end equipment a company production server normally would use.

1.5 Target Audience

This thesis should be of interest to both the individual as well as companies that want to enjoy the security that encryption software can offer while still retaining a fast and responsive system.

[1] http://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0

Everyone that owns a laptop risks having their information compromised if their laptop gets lost or stolen which makes the information in this thesis valuable since it does not only explain what options are available when encrypting the computer but also shows the performance impact when using encryption which is especially crucial for laptops that often have less computing power and slower hard drives.

The theft of laptops and mobile devices is a big problem for companies which the 2008 CSI Computer Crime & Security Survey showed when 42 percent of companies reported to having had either a laptop or a mobile device stolen in the last year alone and that the biggest cost is not the loss of the physical hardware but of the data stored inside [12].

2. Background

2.1 Symmetric Encryption

Symmetric encryption is the encryption type used by the applications that will be benchmarked in this thesis. When using symmetric encryption the same key is used for both the encryption and the decryption of data. Symmetric encryption is also sometimes referred to as “shared-key cryptosystems” because it only makes use of one key for encryption and decryption [13]. Examples of encryption algorithms that use symmetric encryption is AES, Twofish and Serpent [8].

2.2 Encryption Algorithms

This section will explain more about the three different encryption algorithms that will be evaluated in this thesis.

2.2.1 AES (Rijndael)

AES is short for “Advanced Encryption Standard” and is a symmetric encryption algorithm created by Joan Daemen and Vincent Rijmen as a replacement for the DES encryption algorithm. AES is designed as a 128-bit block cipher which can be used with three different cipher key lengths, 128, 192 and 256 bits. In the benchmarks I will be using AES with a 256-bit key length which is the longest and the strongest key length available [13, 14].

2.2.2 Twofish

The symmetric encryption algorithm Twofish is a 128-bit block cipher with a key length of 128, 192 or 256 bits just as the AES algorithm. Twofish was first published in 1998 by the American cryptographer Bruce Schneier [15]. Twofish was created as a replacement for DES and was competing alongside Rijndael, Serpent and twelve other encryption algorithms over which would be the best replacement for DES in the “Advanced Encryption Standard process” contest [4]. Rijndael won the competition and changed its name to AES. Even though Twofish lost it is still used today in some encryption applications such as TrueCrypt, BestCrypt and DiskCryptor.

2.2.3 Serpent

Serpent is a symmetric encryption algorithm that shares the same algorithm specification as AES and Twofish with a 128-bit block cipher and a 128, 192 or 256-bit key length. Serpent was published by Ross Anderson, Eli Biham and Lars Knudsen in 1998 and was a contestant in the “Advanced Encryption Standard process” in which it lost against Rijndael [4, 16]. Just like Twofish it is still being used in a wide range of applications including the three that will be covered in this thesis.

2.3 Encryption Software

Three different encryption applications will be discussed in this chapter and will later be used in the benchmarks.

2.3.1 TrueCrypt

TrueCrypt is a free open-source disk encryption application that runs on Windows, Mac OS X and Linux. TrueCrypt has the ability to create a virtual encrypted disk which is a file that you mount as a real disk. You choose the size of the disk and store data inside this file as if it was a normal disk. This encrypted file can be mounted with a chosen password when the operating system has booted or automatically be mounted upon booting if the operating system partition is encrypted using full system encryption [17].

TrueCrypt has the ability to encrypt a single partition on a hard drive or encrypt a storage device such as a USB thumb drive. TrueCrypt also has the option to perform a full system encryption on the drive where the operating system is installed [17]. When the system encryption option is used you have to provide a password during the operating system boot sequence each time the computer boots and if you are unable to provide the password the operating system will not boot. In order to perform this full system encryption with pre-boot password authentication TrueCrypt uses its own boot loader which is placed on the first track in the boot sector after the initial encryption has taken place [18].

The encryption and decryption of files on the system is performed on-the-fly which means that it is performed in real-time [18]. It is possible that this on-the-fly operation takes extra processing power and reduces the overall performance of the hard drive due to the extra overhead generated but the TrueCrypt team claims that data can be read and written as fast as if the drive was not encrypted due to the pipelining and parallelization technology being used . When parallelization is used on a computer with multiple processors or processor cores TrueCrypt takes advantage of these extra cores during the encryption and decryption of data. Parallelization works by dividing the data being encrypted and decrypted into as many pieces as there are available processors or processor cores on the system and then processing every piece of data with its own processor or processor core in a parallel fashion [19].

To reduce the problem with overhead even further the TrueCrypt developers made use of a technology called “pipelining” or sometimes referred to as “asynchronous processing” which works by decrypting the data in RAM. Decrypting the data in RAM means the application reading the data does not need to wait for the file to be decrypted since RAM is faster than the hard drive. An important thing to note is that Pipelining is only used in the Windows version of TrueCrypt which means other operating systems might run slower [20].

TrueCrypt has the ability to encrypt using three different encryption algorithms [21]. The three algorithms used are AES, Twofish and Serpent and they can be used either separately or in combination with each other which means that if one algorithm gets broken then the data is still protected by the other algorithms [22].

2.3.2 BestCrypt

BestCrypt is a licensed encryption application created by the software developer Jetico. BestCrypt is similar to the functionality of TrueCrypt in that it provides the ability to store files in a container that emulates a hard drive and that can be mounted as a virtual drive. As with TrueCrypt you also need to enter a password every time you mount a container on the system.

BestCrypt gives the user the option to perform a partition encryption or a whole disk encryption. One or multiple partitions can be encrypted while still leaving other partitions on the same hard drive as basic unencrypted partitions as shown in Figure 1.

Whole disk encryption encrypts the entire hard drive regardless of any individual partitions residing on the disk as shown in Figure 2.

The third and final type of disk encryption provided by BestCrypt is called volume encryption. Volume encryption allows encryption on volumes that can span over several disks which is useful when using RAID. As illustrated in Figure 3 the volume encryption can reside on either one disk or span over multiple disks. One volume always use the same password independent of the amount of physical disks used in that volume which is not the case with partition encryption or whole disk encryption where each partition or hard drive should have its own unique password [23].

There is no information on the BestCrypt web page that states if the application uses any method to decrease the overhead created by the encryption such as the pipelining and parallelization technology used by TrueCrypt. A lack of these technologies should manifest a lower encryption and decryption performance rate compared to TrueCrypt and DiskCryptor and if this assumption is accurate or not will be answered in the benchmarks later in this thesis.

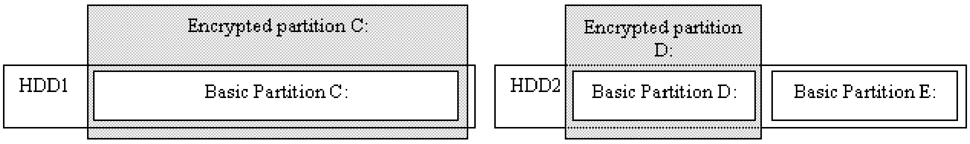


Figure 2: Partition Encryption, Jetico Inc.

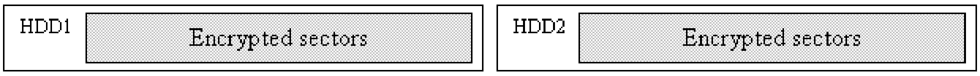


Figure 3: Whole Disk Encryption, Jetico Inc.

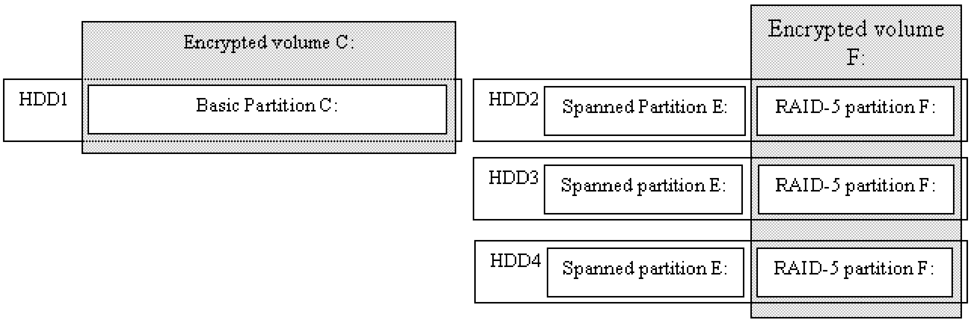


Figure 4: Volume Encryption, Jetico Inc.

2.3.3 DiskCryptor

DiskCryptor is an open source disk encryption application licensed under the GNU General Public License “GNU GPL”. The main features in DiskCryptor is the possibility to encrypt external storage devices such as USB thumb drives as well as the ability to perform encryption on partitions or the entire hard drive. Support for volume encryption also exist which allows for the use of RAID.

The three supported algorithms used by DiskCryptor are AES, Twofish and Serpent. DiskCryptor is similar to TrueCrypt in that the algorithms can be used either separately or in combination with each other in order to achieve a stronger encryption. When using full system encryption DiskCryptor claim to have the performance efficiency of a non-encrypted system which I intend to find out. Similarly to TrueCrypt the application takes advantage of multicore or multiprocessors in order to encrypt and decrypt data in a parallel mode [24].

The support for Intel’s AES hardware acceleration or “AES-SI” is available in TrueCrypt, BestCrypt and DiskCryptor but will not be used in the benchmarks due to the lack of AES-SI support on the Intel Core i5 760 CPU used in the tests [24-27].

2.4 Storage Hardware

Mechanical and solid state drives will be explained in this chapter as they both will be used in the benchmarks.

2.4.1 Mechanical Hard Drive

The mechanical hard drive is a commonly used storage device in computers today. A mechanical hard drive works by using a write/read head that hovers over one or multiple spinning disks in order to access the data that is stored magnetically on these disks. The speed in which a hard drive can spin is measured in RPM which stands for revolutions per minute. The RPM of a hard drive can be as low as 5400 RPM and as high as 15000 RPM depending on the model and what it is being used for. Laptops normally use a 2.5” 5400 RPM hard drive in order to reduce the vibrations and the high power usage needed for a faster hard drive. The main reason for still using mechanical hard drives is that they can store large amounts of data while still keeping a relatively low price compared to that of the solid state drive which provides a vastly better performance but is unable to store as large amount of data and has a considerably higher price tag in comparison to that of the mechanical hard drive [28].

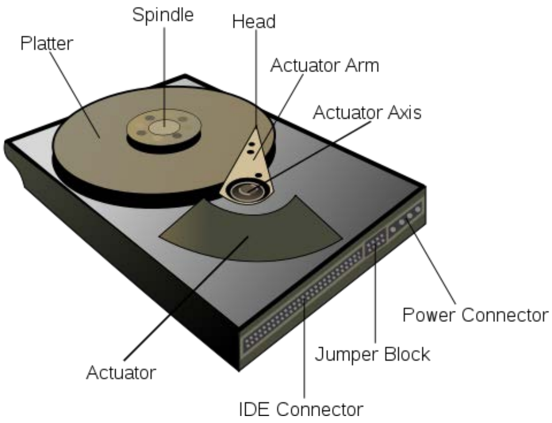


Figure 5: Mechanical Hard Drive, Surachit, 2007

A desktop computer normally uses a 3.5” hard drive with a speed of 7200 or 10000 RPM. This high speed and larger area provides a faster read and write access time and more storage space but uses up more electricity and risks creating more vibrations. In this thesis I will perform benchmarks on 5400 RPM, 7200 RPM and 10000 RPM hard drives. In figure 4 you can see the parts that are included in a typical hard drive [29].

2.4.2 Solid State Drive

A solid state drive differ from a mechanical hard drive in that it has no moving parts and instead makes use of non-volatile flash memory to store the data. Non-volatile means that the data is not lost from the memory chip when the power is lost. An SSD can be found in either a 2.5” casing or as a PCI-card. The performance difference between a solid state drive and a physical hard drive is very big since the solid state drive does not need any moving parts to read its data [28].

3. Method

This section will explain in detail how the encryption applications were installed and how to encrypt the storage devices as well as present the benchmarking tools used.

3.1 Scientific Approach

In this thesis I chose to use deductive reasoning and quantitative research in order to gather unbiased statistics through empirical observation. In order to gather this information a set of benchmark tools were used. Since the same sets of tools were used for all benchmarks under the same circumstances the results are scientific and impartial.

3.2 Experiment

All the experiments were performed on a Windows 7 Ultimate 64-bit operating system. The reason Windows 7 will replace Windows XP is that Microsoft has stopped giving support for it and because of the hardware limitations set on Windows XP because of its 32-bit architecture more people will decide to upgrade to Windows 7 64-bit in order to be able to take advantage of the increased memory allocation and to be able to run 64-bit applications. Using 64-bit Windows 7 as the operating system for the experiments means the results will be able to reach a wide audience and be more relevant in the future when Windows XP drops in usage. Because all the applications tested in this thesis do support Windows XP it is possible that the results could be applicable for Windows XP as well [11].

Before any benchmarks were performed on the PC all the applications that were not needed for the experiment were turned off so that they would not affect the results.

A baseline performance benchmark was created by running the benchmarking tools on the storage devices before any encryption had taken place. These baseline results were used in the performance charts under the title “Unencrypted” and were used to compare the different encryption applications and algorithms.

Motherboard	ASRock P55 Extreme
Processor	Intel Core i5 760 running at 2.81 GHz
GPU	NVIDIA GeForce GTX 680
Memory	Corsair 8GB (4x2048MB) 1600MHz XMS3
Power Supply	Corsair HX 750W 80+
Hard Drive 1	Intel 320 Series Solid State Drive SSDSA2CW120G3 Internal 120GB

	SATA II 2.5’: 6.1.7600.16385
Hard Drive 2	Seagate Barracuda 7200RPM 32MB 500GB SATA II 3.5’: 6.1.7600.16385
Hard Drive 3	Western Digital Scorpio Blue 8MB 400GB SATA II 2.5’
Hard Drive 4	Western Digital Raptor X WD1500AHFD Gamer 10000RPM 16MB 150 GB, SATA II 3.5’

Table 1: Hardware specifications

The storage devices used in the benchmarks comprise of one solid state drive and three mechanical hard drives and their specifications are displayed in the list above. All storage devices were connected to a SATA II interface on the motherboard. The hardware specifications of the PC components used during the benchmarking is displayed in Table 1.

Encrypting large storage devices takes a lot of time and for this reason a 10GB partition was created on the storage devices which reduced the encryption time from a couple hours to approximately 10 minutes per algorithm.

3.2.1 TrueCrypt Installation

TrueCrypt was downloaded from the TrueCrypt webpage and then installed on the PC. In order to encrypt the desired partition I left clicked on the “Volumes” button in the upper left corner and then chose “Create New Volume” which initiated the “TrueCrypt Volume Creation Wizard”.

Under this wizard the “Encrypt a non-system partition/drive” option was chosen and then the “Next” button was pressed to continue to the next page where the “Standard TrueCrypt volume” option was chosen. After pressing the “Next” button the wizard prompted for a volume location. The volume location was selected by pressing the “Select Device” button and then choosing the desired partition in the list. The next page displayed the “Volume Creation Mode” where the “Create encrypted volume and format it” option was chosen.

After pressing the “Next” button the options for the desired encryption algorithm and hash algorithm was displayed. The encryption algorithm was chosen from the dropdown list starting with AES. The “SHA-512” hash algorithm was chosen and then the “Next” button was pressed twice to proceed to the “Volume Password” menu where a password was entered. After pressing “Next” the wizard asked if large files were going to be stored on the volume. The “Yes” option was chosen enabling NTFS and then the “Next” button was pressed to continue to the last step in the wizard. In order to create a random key pool the mouse was moved in a random pattern for 10 seconds and then the “Format” button was pressed which started the encryption process.

When the encryption of the volume was completed the volume was mounted by pressing the “Select Device…” button which displayed a list of available hard drives and partitions. The encrypted volume was chosen and the “OK” button was pressed which opened a prompt for the volume password. The password was entered and then volume was mounted. Now the encrypted volume was available as a hard drive under the “Computer” icon in Windows 7. The benchmarking tools were placed onto the volume and the benchmarking commenced.

After the benchmarking was completed and the results collected the volume creation wizard was initiated again in order to select a new encryption algorithm. This was done for all the encryption algorithms on all the storage devices.

3.2.2 DiskCryptor Installation

DiskCryptor was downloaded from diskcryptor.net and then installed on the PC. In order to encryption the volume the application was started and the desired partition was chosen from the list in the main menu. After selecting the partition the left button was pressed which opened a list of options where the “Encrypt” option was chosen. The encryption algorithm was chosen starting with AES and then the “Next” button was pressed to continue. The volume password was entered and then the “OK” button was pressed in order to start the encryption process. After finishing the encryption process the volume was automatically mounted and then the benchmarking tools were placed onto the encrypted volume and after that the testing began. The “Reencrypt” option was used to encrypt the volume with the remaining algorithms. This was performed on all the storage devices.

3.2.3 BestCrypt Installation

BestCrypt was downloaded from Jetico.com and then installed on the PC. In order to encrypt a volume the desired partition was selected in the partition list and then the “Encrypt Volume” option was chosen. After this the encryption algorithm was chosen starting with AES. After choosing the encryption algorithm a password was entered and then the “OK” button was selected in order to start the encryption process.

After the volume was encrypted it was automatically mounted. The benchmarking tools were placed on the volume and the testing was started. This was done for all algorithms on all storage devices. As BestCrypt does not support the combination of algorithms such as AES-Twofish and AES-Twofish-Serpent these benchmarks could not be performed.

3.3 Benchmarking Tools

The two benchmarking tools used were Anvil’s Storage Utilities 1.0.34 Beta11 and CrystalDiskMark 3.0.1. Both these applications perform a variety of different disk operations in order to simulate the conditions that a storage device can be exposed to. The data collected from the benchmarks was used to create performance charts that are presented in the results chapter.

3.3.1 Anvil's Storage Utilities

Anvil's Storage Utilities is a benchmarking application that can be used for both mechanical hard drives and solid state drives.

The default settings were used during all the benchmarks. The default settings include the following disk operations. Sequential Read 4MB, Read 4K, Read 4K QD4, Read 4K QD16, Read 32K, Read 128K, Sequential Write 4MB, Write 4K, Write 4K QD4 and Write 4k QD16. The data collected from these disk operations was the response time, the amount of data read or written in megabytes, the input/output operations per second and the speed in megabytes per second.

The read operations were collected into a final read score, the write operations were collected into a final write score and based on all results a total score was calculated by the benchmarking application. When starting the benchmark in Anvil's Storage Utilities a 1GB file is created on the selected storage device which is used during the benchmarking of the storage device [30].

3.3.2 CrystalDiskMark

CrystalDiskMark is a storage benchmarking application. CrystalDiskMark performs the following disk operations. Sequential Read, Sequential Write, Random Read 512KB, Random Write 512KB, Random Read 4 KB QD1, Random Write 4KB QD1, Random Read 4KB QD32 and Random Write 4KB QD32.

When starting a benchmark CrystalDiskMark creates a 1GB test file on the storage device currently selected which it uses to perform the read and write operations. In order to increase the reliability CrystalDiskMark performs every disk operation five times and then calculate the average value which is then displayed [31].

3.4 Method Discussion

The reason for using Anvil’s Storage Utilities and CrystalDiskMark and not the other eight benchmarking tools that also were evaluated is that they both are very easy to use and perform the benchmarks in a timely manner and also present the results in such a way that it became very easy to create charts that are easy to understand.

The tool that provided the best overview of the performance was Anvil’s Storage Utilities since it took a wide number of different factors and created a single read, write and total score which made it easy to create charts that displayed the performance in a simple graphical illustration compared to CrystalDiskMark which instead showed the performance of the various disk operations individually. This method of presenting the data is however also valuable in order to be able to deduce the underlying reason why certain applications and algorithms perform in certain ways.

4. Results

In this chapter the data collected from the benchmarks is presented in charts that display the performance for the different encryption applications depending on the encryption algorithm that was used and on which type of storage device the benchmark was performed on.

4.1 Observation: 01 SSD Charts

The following charts were performed on the solid state drive. The combination of multiple algorithms such as AES-Twofish and AES-Twofish-Serpent is not supported by BestCrypt and is therefore only included for TrueCrypt and DiskCryptor in the comparison charts. Especial consideration should be taken to figure 12 that creates a summary of all the total score results from the Anvil's Storage Utilities across the different applications

Figure 6 shows the read score, the write score and the total score on the SSD when using the encryption application TrueCrypt. When comparing the total unencrypted score with the total score of the fastest algorithm which is Twofish, there is a 40.5% decrease in performance. Chart shows performance in read, write and total score. Higher score is better. Table 2 shows the results in detail.

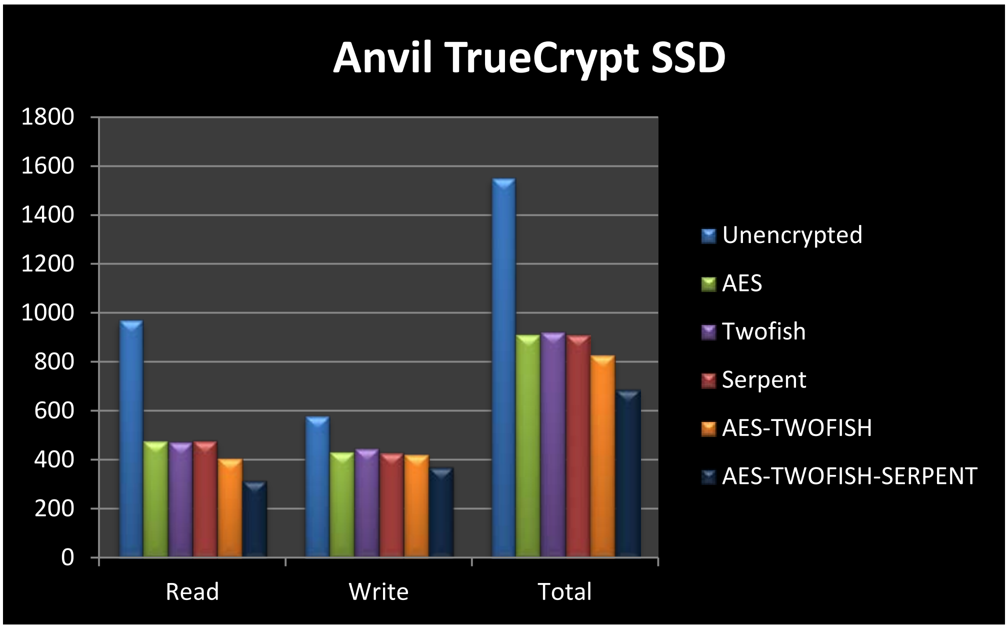


Figure 6: Anvil's Storage Utilities TrueCrypt SSD

TrueCrypt Anvil SSD	Read	Write	Total
Unencrypted	971,29	581,29	1552,59
AES	480,31	432,01	912,33
Twofish	472,7	450,05	922,75
Serpent	480,31	432,01	912,33
AES-TWOFISH	406,88	424,34	831,22
AES-TWOFISH-SERPENT	316,28	371,73	688,02

Table 2: Anvil's Storage Utilities TrueCrypt SSD results

Figure 7 shows the speed in MB/s for the different disk operations and encryption algorithms when using the TrueCrypt application on the SSD. Unencrypted performance is generally higher across the different disk operations except under the “Sequential Write” and “Random Write 512KB” where the performance is unsubstantially lower. Chart shows performance in megabytes per second across the different disk operations and encryption algorithms. Higher score is better. Table 3 shows the results in detail.

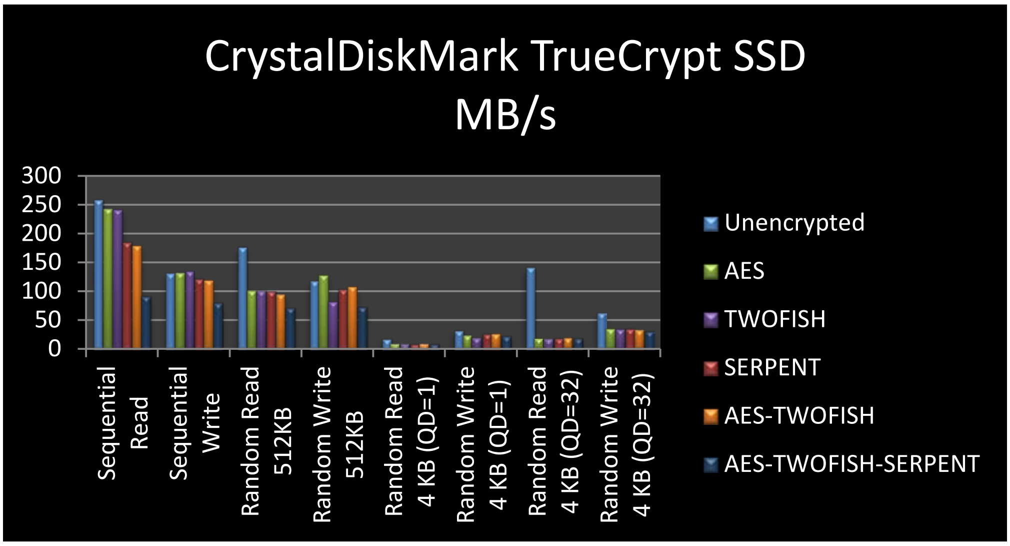


Figure 7: CrystalDiskMark TrueCrypt SSD.

TrueCrypt Crystal SSD	Sequential Read	Sequential Write	Random Read 512KB	Random Write 512KB
Unencrypted	259	132	177	119
AES	244,7	133,3	101,3	127,9
Twofish	241,7	134,5	102	81,8
Serpent	184,8	121,3	100,2	103,5
AES-TWOFISH	179,3	119,3	96,28	107,8
AES-TWOFISH-SERPENT	90,97	79,76	70,8	72,75
	Random Read 4KB QD1	**Random Write 4KB QD1**	**Random Read 4KB QD32**	**Random Write 4KB QD32**
Unencrypted	19	33	142	64
AES	9,511	24,35	19	34,94
Twofish	9,481	20,78	18,57	34,5
Serpent	9,646	27,46	19,62	36,79
AES-TWOFISH	9,476	26,12	19,65	33,84
AES-TWOFISH-SERPENT	9,839	23,94	19,19	32,1

Table 3: CrystalDiskMark TrueCrypt SSD.

Figure 8 shows the read score, the write score and the total score on the SSD when using the application DiskCryptor. As seen in the figure the read and write performance of the Twofish algorithm is marginally faster than AES with a total performance increase of 1.8%. When comparing the score of the fastest algorithm which is Twofish with the unencrypted score there is a 16.3% decrease in performance. Chart shows performance in read, write and total score. Higher score is better. Table 4 shows the results in more detail.

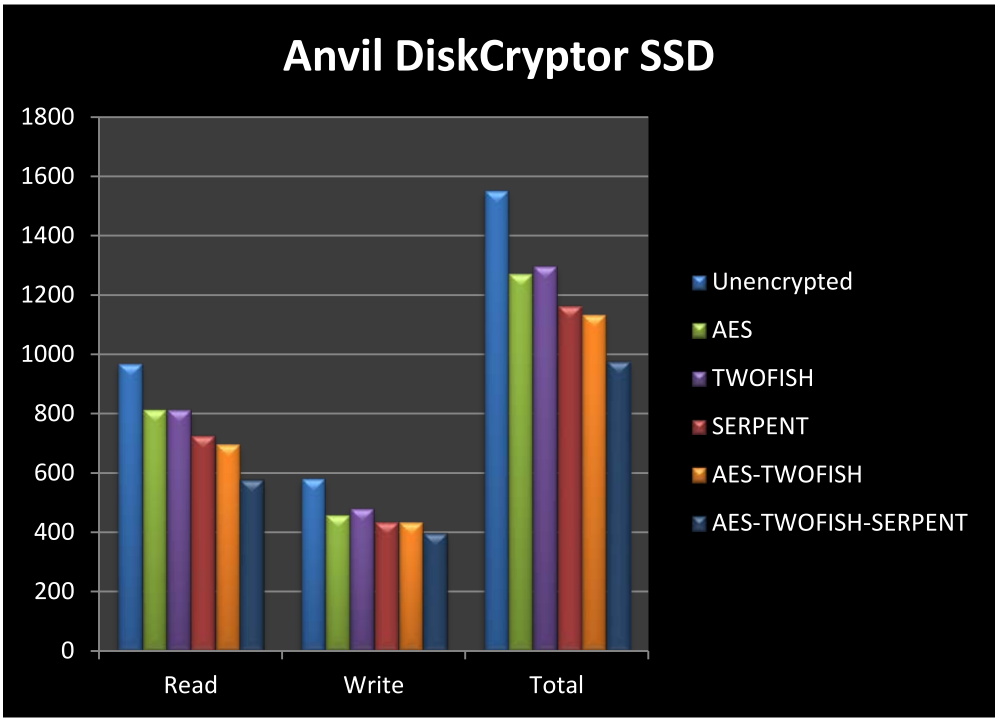


Figure 8: Anvil's Storage Utilities DiskCryptor SSD.

DiskCryptor Anvil SSD	Read	Write	Total
Unencrypted	971,29	581,29	1552,59
AES	814,69	459,97	1274,66
Twofish	817,32	481,02	1298,34
Serpent	726,99	438,07	1165,06
AES-TWOFISH	699,21	437,38	1136,58
AES-TWOFISH-SERPENT	579,69	399,24	976,93

Table 4: Anvil's Storage Utilities DiskCryptor SSD.

Figure 9 shows the speed in MB/s for the different algorithms when running the application DiskCryptor on the SSD. DiskCryptor is substantially faster than both TrueCrypt and BestCrypt when looking at the “Random Read 4KB” and “Random Write 4 KB”. Chart shows performance in megabytes per second across the different disk operations and encryption algorithms. Higher score is better. Table 5 shows the results in further detail.

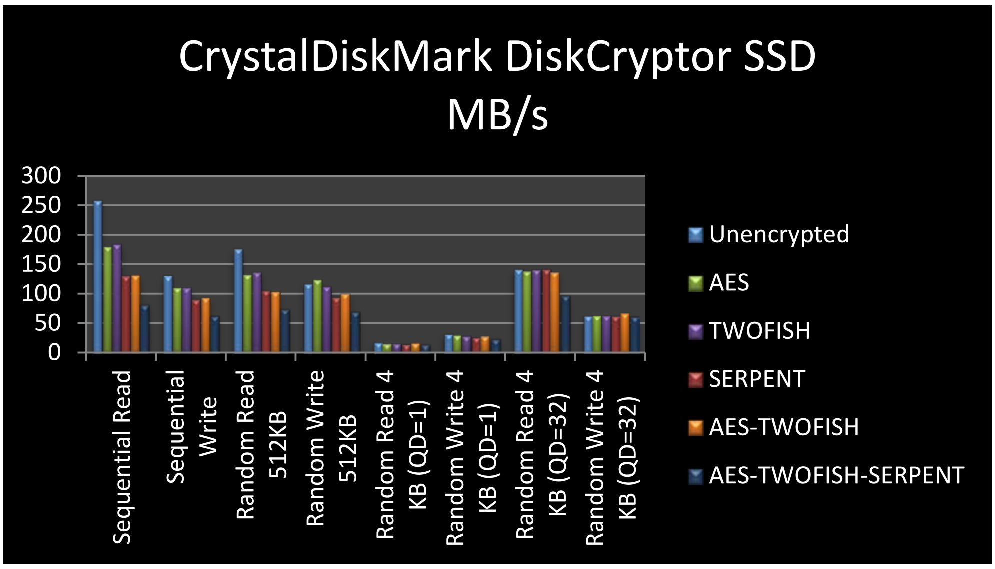


Figure 9: CrystalDiskMark DiskCryptor SSD.

DiskCryptor Crystal SSD	Sequential Read	Sequential Write	Random Read 512KB	Random Write 512KB
Unencrypted	259	132	177	119
AES	180,3	110,9	133,3	123,5
Twofish	184,4	111,2	137,4	112,3
Serpent	131	90,55	105,6	94,29
AES-TWOFISH	131,7	93,34	103,5	100,9
AES-TWOFISH-SERPENT	81,05	62,42	73,54	69,37
	Random Read 4KB QD1	Random Write 4KB QD1	Random Read 4KB QD32	Random Write 4KB QD32
Unencrypted	19	33	142	64
AES	16	29,38	139,1	62,68
Twofish	15,99	27,69	141,9	63,63
Serpent	15,79	27,25	141,5	63,41
AES-TWOFISH	16,48	28,65	137	67,49
AES-TWOFISH-SERPENT	15,15	24,45	96,52	61,32

Table 5: CrystalDiskMark DiskCryptor SSD.

Figure 10 shows the read score, the write score and the total score on the SSD when using the application BestCrypt. When comparing the score of the fastest algorithm which is Twofish with the unencrypted score there is a 47.8% decrease in performance. Chart shows performance in read, write and total score. Higher score is better. Table 6 shows the results in more detail.

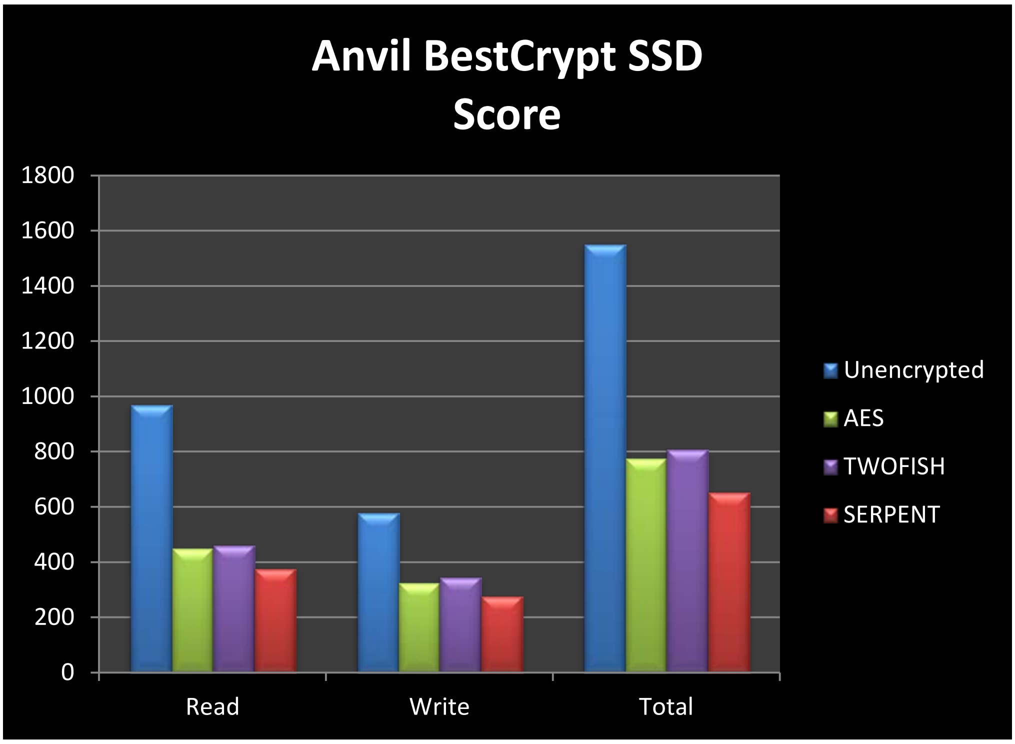


Figure 10: Anvil's Storage Utilities BestCrypt SSD.

BestCrypt Anvil SSD	Read	Write	Total
Unencrypted	971,29	581,29	1552,59
AES	450,82	327,39	778,22
Twofish	462,92	347,14	810,06
Serpent	377	276,88	654,48

Table 6: Anvil's Storage Utilities BestCrypt SSD.

Figure 11 shows the speed in MB/s for the different algorithms when running the application BestCrypt on the SSD. The Twofish algorithm has the best performance during all the disk operations. Chart shows performance in megabytes per second across the different disk operations and encryption algorithms. Higher score is better. Table 7 shows the results in more detail.

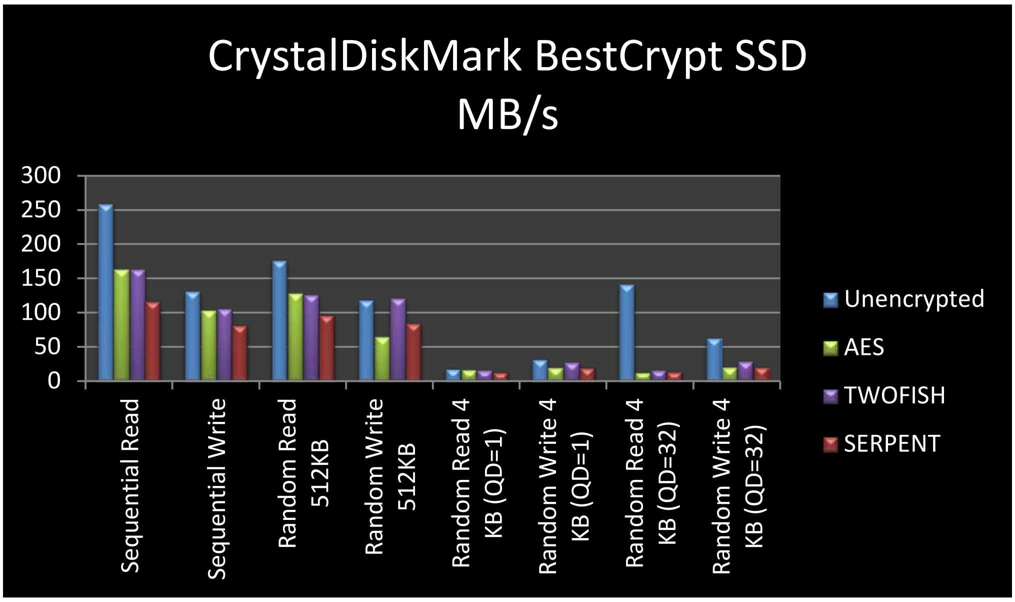


Figure 11: CrystalDiskMark BestCrypt SSD.

BestCrypt Crystal SSD	Sequential Read	Sequential Write	Random Read 512KB	Random Write 512KB
Unencrypted	259	132	177	119
AES	164,6	103,6	128,3	65,44
Twofish	164,4	105,7	127,3	121,4
Serpent	116,5	81,88	96,47	84,48
	Random Read 4KB QD1	**Random Write 4KB QD1**	**Random Read 4KB QD32**	**Random Write 4KB QD32**
Unencrypted	19	33	142	64
AES	16	20,4	12,36	20,66
Twofish	16,49	28,05	16,55	29,61
Serpent	14,27	20,69	14,58	21,63

Table 7: CrystalDiskMark BestCrypt SSD.

Figure 12 shows the total Anvil score for all the encryption applications and the supported algorithms. The performance across all algorithms is higher when using DiskCryptor compared to both TrueCrypt and BestCrypt. When looking at the multi-layer algorithms there is a substantial performance decrease when using TrueCrypt compared to DiskCryptor with a total performance decrease of 26.8% for AES-Twofish and 29.7% for AES-Twofish-Serpent. Table 8 shows the results in more detail.

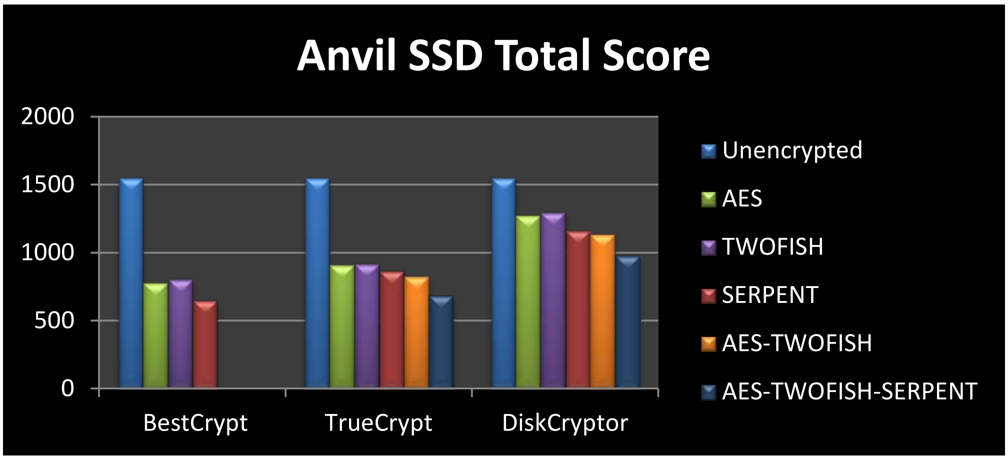
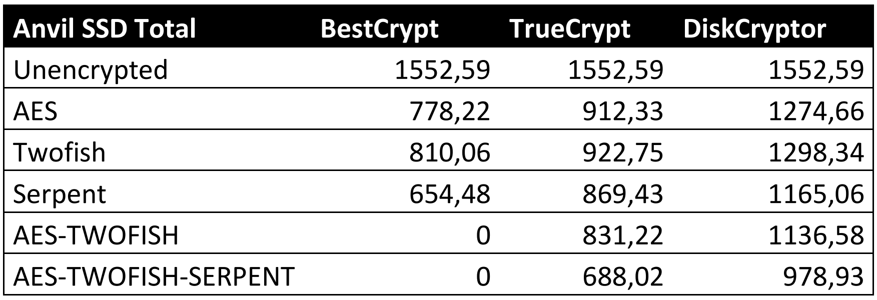


Figure 12: Anvil's Storage Utilities benchmark comparing the total score for all algorithms and all encryption applications on a SSD.



Anvil SSD Total	BestCrypt	TrueCrypt	DiskCryptor
Unencrypted	1552,59	1552,59	1552,59
AES	778,22	912,33	1274,66
Twofish	810,06	922,75	1298,34
Serpent	654,48	869,43	1165,06
AES-TWOFISH	0	831,22	1136,58
AES-TWOFISH-SERPENT	0	688,02	978,93

Table 8: Anvil's Storage Utilities benchmark comparing the total score for all algorithms and all encryption applications on a SSD.

4.2 Observation: 02 Mechanical Hard Drive Charts

The following charts were performed on the mechanical hard drives. The combination of multiple algorithms such as AES-Twofish and AES-Twofish-Serpent is not supported by BestCrypt and is therefore only included for TrueCrypt and DiskCryptor in the comparison charts. The performance pattern for the CrystalDiskMark benchmarks is very similar across the 7200 RPM, 10000 RPM and 5400 RPM hard drives and will for this reason only be included for the 7200 RPM hard drive. The performance pattern across the different hard drives is also present in the Anvil benchmarks and because of this similarity only the benchmarks with the total Anvil score will be displayed in order to provide good overview.

Figure 13 shows the speed in MB/s for the different algorithms when running TrueCrypt on a mechanical hard drive. TrueCrypt manages to keep a steady performance across all the disk operations and algorithms except during the “Random Write 512KB” where there is a ≈20% decline compared to the unencrypted performance. Chart shows performance in megabytes per second across the different disk operations and encryption algorithms. Higher score is better. More detailed information can be found in table 9.

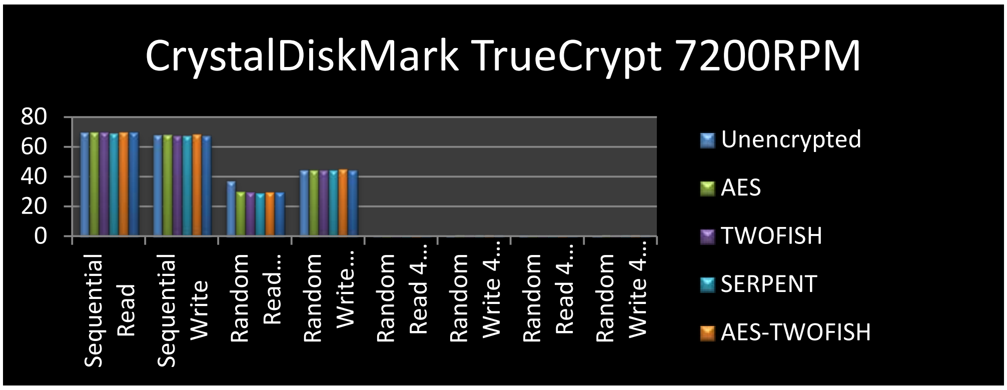


Figure 13: CrystalDiskMark TrueCrypt Mechanical Hard Drive.

TrueCrypt Crystal 7200RPM	Sequential Read	Sequential Write	Random Read 512KB	Random Write 512KB
Unencrypted	70,14	68,44	37,7	44,93
AES	70,13	68,7	30,33	44,5
Twofish	60,99	68,13	30,15	44,74
Serpent	69,78	68,17	29,59	44,56
AES-TWOFISH	70,02	68,5	29,9	45,06
AES-TWOFISH-SERPENT	69,98	68,05	29,75	44,82
	Random Read 4KB QD1	**Random Write 4KB QD1**	**Random Read 4KB QD32**	**Random Write 4KB QD32**
Unencrypted	0,644	1,19	1,681	1,196
AES	0,579	1,193	0,578	1,193
Twofish	0,584	1,195	0,651	1,193
Serpent	0,588	1,238	0,591	1,204
AES-TWOFISH	0,568	1,204	0,592	1,216
AES-TWOFISH-SERPENT	0,586	1,227	0,59	1,208

Table 9: CrystalDiskMark TrueCrypt Mechanical Hard Drive.

Figure 14 shows the speed in MB/s for the different algorithms when running DiskCryptor on a mechanical hard drive. The results are very similar to those in Figure 13 with a steady performance across the different disk operations and algorithms except during the "Random Read 512KB" where the Serpent and AES-Twofish-Serpent displays a marginal performance decrease. Chart shows performance in megabytes per second across the different disk operations and encryption algorithms. Higher score is better. Detailed information about the graph can be found in table 10.

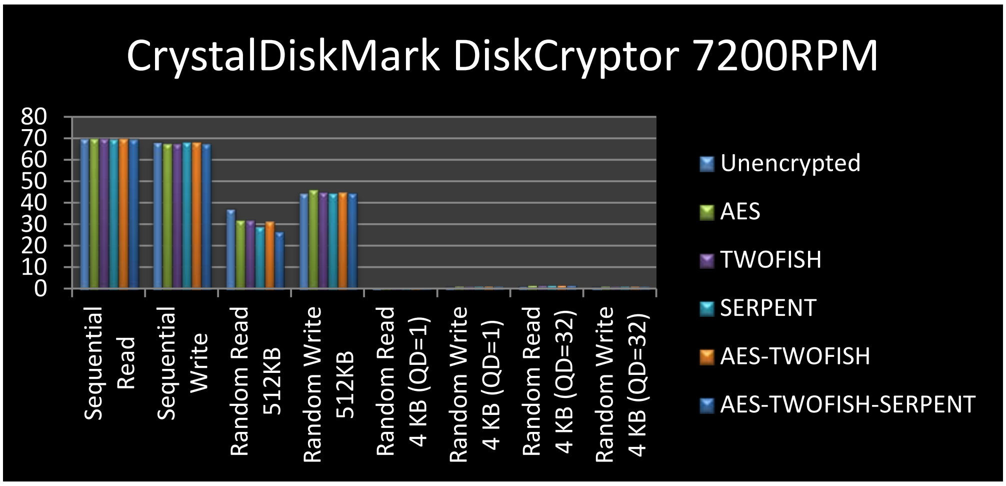


Figure 14: CrystalDiskMark DiskCryptor Mechanical Hard Drive.

DiskCryptor Crystal 7200RPM	Sequential Read	Sequential Write	Random Read 512KB	Random Write 512KB
Unencrypted	70,14	68,44	37,7	44,93
AES	70,1	67,91	31,98	46,23
Twofish	70	67,91	32,15	45,05
Serpent	69,98	68,24	29,27	44,59
AES-TWOFISH	70,08	68,46	31,73	45,34
AES-TWOFISH-SERPENT	69,91	68,02	26,5	44,63
	Random Read 4KB QD1	**Random Write 4KB QD1**	**Random Read 4KB QD32**	**Random Write 4KB QD32**
Unencrypted	0,644	1,19	1,681	1,196
AES	0,651	1,189	1,731	1,178
Twofish	0,649	1,172	1,712	1,171
Serpent	0,65	1,191	1,723	1,196
AES-TWOFISH	0,634	1,213	1,718	1,172
AES-TWOFISH-SERPENT	0,642	1,19	1,698	1,173

Table 10: CrystalDiskMark DiskCryptor Mechanical Hard Drive.

Figure 15 shows the speed in MB/s for the different algorithms when running the BestCrypt application on a mechanical hard drive. As in Figure 13 and 14 the results are very similar when it comes to performance between the different algorithms. Chart shows performance in megabytes per second across the different disk operations and encryption algorithms. Higher score is better. More detailed information can be found in table 11.

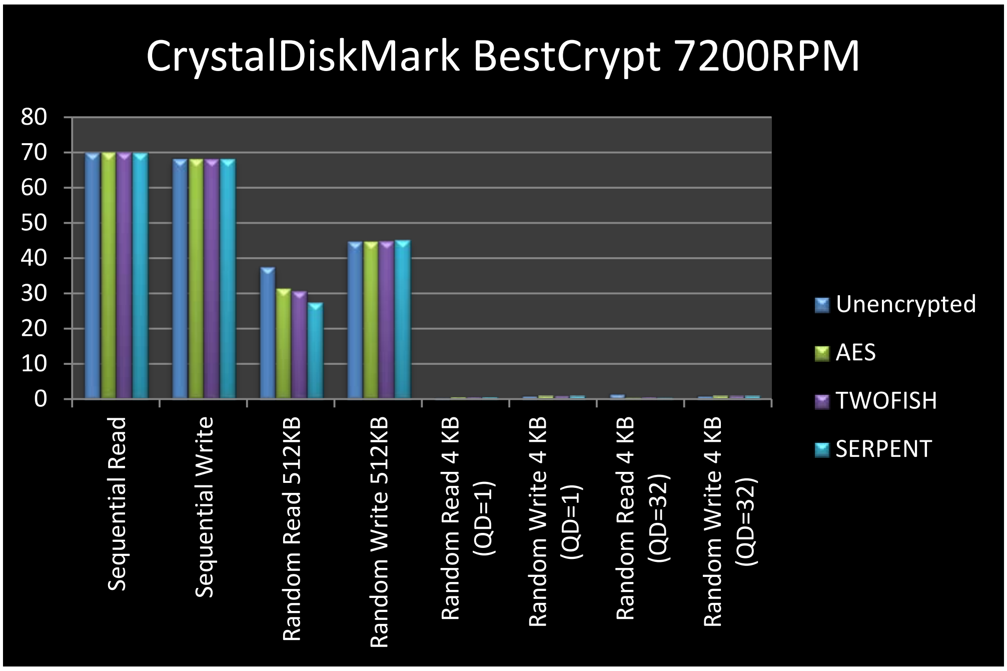


Figure 15: CrystalDiskMark BestCrypt Mechanical Hard Drive.

BestCrypt Crystal 7200RPM	Sequential Read	Sequential Write	Random Read 512KB	Random Write 512KB
Unencrypted	70,14	68,44	37,7	44,93
AES	70,09	68,28	31,48	44,88
Twofish	70,13	68,28	31,03	45,17
Serpent	70,02	68,24	27,65	45,47
	Random Read 4KB QD1	**Random Write 4KB QD1**	**Random Read 4KB QD32**	**Random Write 4KB QD32**
Unencrypted	0,644	1,19	1,681	1,196
AES	0,63	1,198	0,57	1,187
Twofish	0,633	1,214	0,612	1,212
Serpent	0,623	1,211	0,57	1,185

Table 11: CrystalDiskMark BestCrypt Mechanical Hard Drive.

Figure 16 shows the total Anvil score for all the encryption applications and supported algorithms when using a 7200 RPM mechanical hard drive. There is a difference in performance between the different applications and the AES, Twofish and Serpent algorithms but not large enough to be statistically significant. There is however a noticeable difference in performance when comparing TrueCrypt and DiskCryptor in the implementation of multi-layer encryption where DiskCryptor is 7.9% slower when using AES-Twofish and 25.1% slower when using AES-Twofish-Serpent. The exact results from Figure 16 can be found in table 12.

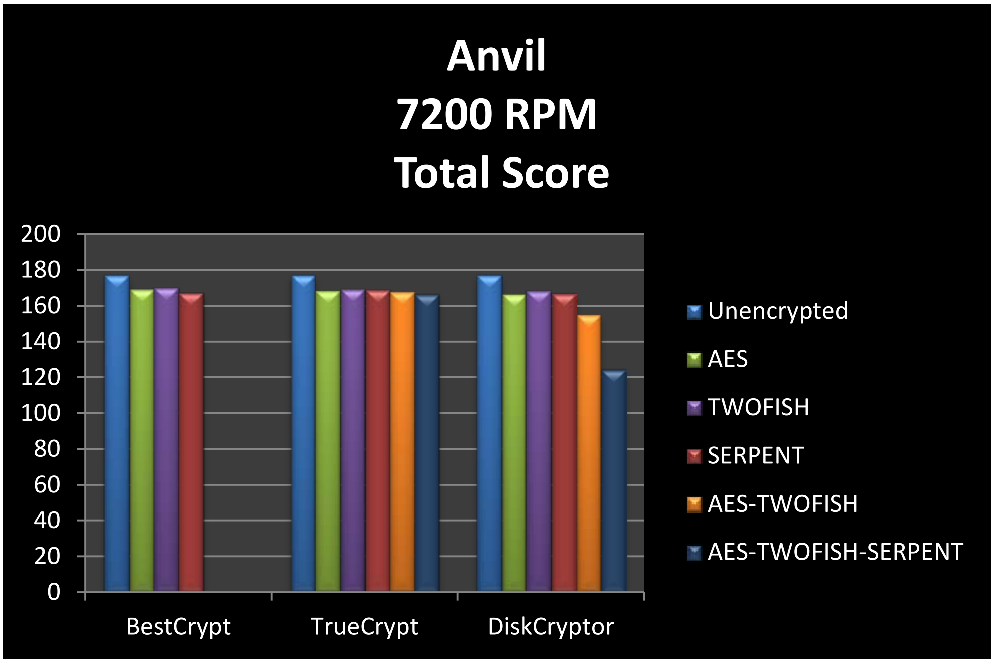


Figure 16: Anvil's Storage Utilities Total Score 7200RPM.

Anvil 7200RPM Total	BestCrypt	TrueCrypt	DiskCryptor
Unencrypted	177,29	177,29	177,29
AES	169,62	168,72	166,53
Twofish	169,99	169,49	168,8
Serpent	167,48	169	167
AES-TWOFISH	0	168,33	155,01
AES-TWOFISH-SERPENT	0	166,51	124,62

Table 12: Anvil's Storage Utilities Total Score 7200RPM.

Figure 17 shows the total Anvil score for all the encryption applications and the supported algorithms when using a 10000 RPM mechanical hard drive. As seen in the figure TrueCrypt manages to keep the performance even across all the algorithms compared to BestCrypt and DiskCryptor where you can see a ≈8.8% decline in performance for the Serpent algorithm. When looking at the multi-layer algorithms there is a performance decrease of 6.5% for AES-Twofish and 12.3% for AES-Twofish-Serpent when using DiskCryptor compared to TrueCrypt. More detailed information about the graph can be found in table 13.

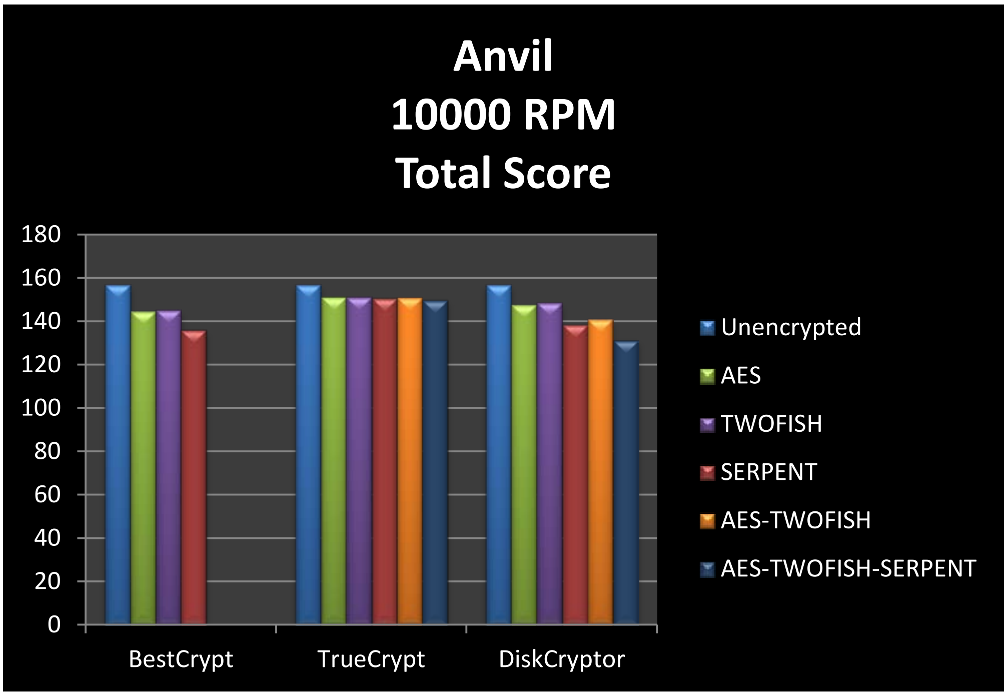


Figure 17: Anvil's Storage Utilities Total Score 10000RPM

Anvil 10000RPM Total	BestCrypt	TrueCrypt	DiskCryptor
Unencrypted	157,23	157,23	157,23
AES	144,87	151,09	147,93
Twofish	145,25	151,2	148,71
Serpent	136,33	150,9	138,73
AES-TWOFISH	0	151,05	141,17
AES-TWOFISH-SERPENT	0	150,07	131,55

Table 13: Anvil's Storage Utilities Total Score 10000RPM

Figure 18 shows the total Anvil score for all the encryption applications and the supported algorithms when using a 5400 RPM mechanical hard drive. As seen in the figure TrueCrypt once again manages to keep the performance even across all the algorithms compared to both BestCrypt and DiskCryptor where there is a noticeable decline in performance for the Serpent algorithm. When looking at the multi-layer algorithms there is a performance decrease of 4.9% for AES-Twofish and 21.3% for AES-Twofish-Serpent when using DiskCryptor compared to TrueCrypt.

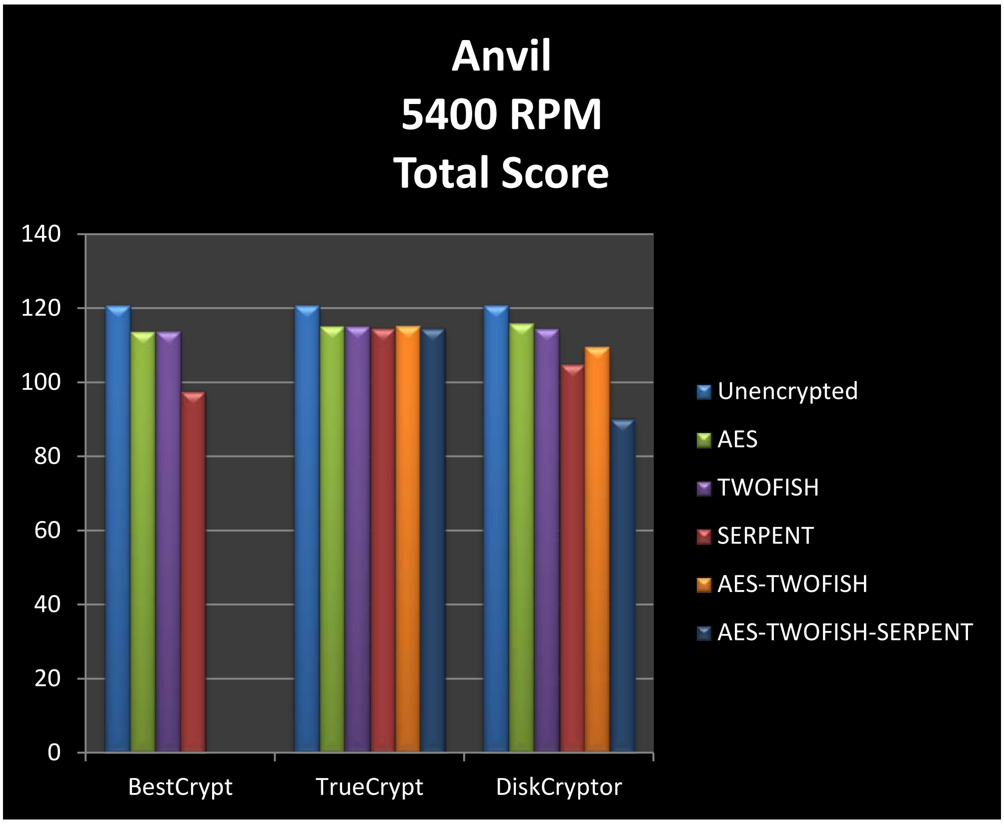


Figure 18: Anvil's Storage Utilities Total Score 5400RPM.

Anvil 5400RPM Total	BestCrypt	TrueCrypt	DiskCryptor
Unencrypted	121,08	121,08	121,08
AES	113,87	115,28	116,24
Twofish	114,03	115,25	114,55
Serpent	97,74	114,75	105,06
AES-TWOFISH	0	115,54	109,85
AES-TWOFISH-SERPENT	0	114,78	90,31

Table 14: Anvil's Storage Utilities Total Score 5400RPM.

5. Discussion

In this chapter there will be an analysis of the results and a discussion about the significance and usefulness of the findings made in this thesis. A discussion about the reliability and validity will also be covered in this chapter under methodology reflection.

5.1 Discussion and analysis of results

The difference in performance when using a solid state drive was surprisingly substantial which can be observed in Figure 12 where the difference in implementation of the Twofish algorithm resulted in a staggering 37.6% performance decrease when using BestCrypt compared to DiskCryptor and a decrease of 28.9% when using TrueCrypt compared to DiskCryptor.

This immense difference in performance is not restricted to just Twofish as the difference in the implementation of AES showed that BestCrypt had a performance decrease of 38.9% when comparing it to DiskCryptor and that TrueCrypt suffered a performance decrease of 28.4% when juxtaposed with DiskCryptor. For the Serpent algorithm the decrease in performance was 37.5% for BestCrypt and 28.9% for TrueCrypt compared to that of DiskCryptor.

One possible explanation for this substantial difference in performance can be observed in Figure 9 where it would appear that DiskCryptor somehow manages to sustain a high performance during random read and write operations that have a high queue depth even when using encryption which is not the case with either TrueCrypt or BestCrypt. It is unfortunate that the lack of resources made it impossible to benchmark a wider array of different solid state drives. This lack of additional solid state drives makes it difficult to determine if the results are connected to the specific Intel 320 Series SSD that was used during the benchmarks and that it therefore might be an isolated event that could be related to problems in the firmware.

Even though the performance gap seen between the different encryption applications when using a solid state drive is not as large when looking at the performance charts for the mechanical hard drives there still is a noticeable difference.

When looking at the Anvil total score for the mechanical hard drives it becomes clear that TrueCrypt is the application that best manages to keep a steady performance across all the algorithms which could be because of the parallelization and pipelining technology that it has implemented.

It is quite remarkable that the performance increase that DiskCryptor displayed when using a solid state drive does not present itself when using a mechanical hard drive but instead displays a lower performance across all the algorithms on all the mechanical hard drives when comparing it to TrueCrypt. Even though BestCrypt manifested a low performance on the solid state drive it does manages to redeem itself as it is the application with the best results on the 7200 RPM hard drive. Unfortunately for BestCrypt this performance increase is not presented on the other two hard drives.

By looking at the results from the SSD benchmarks it becomes apparent that choosing the right encryption application is crucial in order to reduce the negative performance impact that encryption produces. It also becomes apparent by looking at the BestCrypt results for the SSD benchmarks that paying for your encryption software does not necessarily mean that you get better performance but instead quite the contrary as it displayed the worst performance across all the supported algorithms.

When looking at the performance difference shown in the solid state drive charts it becomes difficult to determine how this difference would affect the end-user without using the system for an extended period of time and then evaluating the overall performance experienced. The difference is however substantial enough to not be overlooked. Rating the end-user experience becomes an even more daunting task when looking at the differences shown in the mechanical hard drive charts. There could however be a significant impact on systems that have a high I/O workload such as a database server.

5.2 Methodology Reflection

By using two different benchmarking applications that both measured a wide variety of different disk operations on the storage devices should have resulted in a high validity for the benchmarks. The reliability was also high when looking at the actual benchmarks performed since Anvil's Storage Utilities perform every disk operation for a period of fifteen seconds and then collects the average and CrystalDiskMark perform every disk operation five times before saving the average.

It is possible that the reliability could be even higher if a longer period of time and even more repetitions were used but this would have taken too long to be feasible with the restricted time period set. The decision to use 10GB partitions on the mechanical hard drives is also a factor that could affect the reliability in a negative way since the placement of the partition on the hard drive is unknown which means the partitions might be placed further out on the disk area which reduces the performance of the partition compared to that of the actual drive.

The consequences of this decision can be seen in all the mechanical hard drive charts where the overall performance is lower than that of the actual hard drive. The actual significance of the results should not be affected since the performance decrease occurred on all the drives which resulted in the 7200 RPM hard drive being faster than the 10000 RPM hard drive and the 5400 RPM hard drive being slower than both the 7200 RPM and 10000 RPM hard drive which is accurate.

Since the solid state drive does not use any mechanical parts it was not affected by the decision to use partitions.

The reason for using a 10000 RPM hard drive was to see if there would be any difference in in performance in the benchmarks when using a drive with very high performance. This idea was ruined when it was discovered that the old age of the hard drive meant that even though it had a faster rotational speed it was actually slower than the 7200 RPM hard drive but faster than the 5400 RPM drive.

The last factor relates to the external validity of the SSD results. Since only one solid state drive was available for benchmarking on the question of whether the results are generalizable or not is difficult to determine without repeating the experiments on other solid state drives.

6. Conclusion

This chapter will consist of a presentation of the final thoughts and conclusions regarding the purpose of this thesis and if the questions posed in the problem description in chapter 1.3 were answered in a satisfactory manner. There will also be a discussion regarding proposals for future research in the area.

6.1 Final words and conclusions

The purpose of this thesis was to evaluate three encryption applications and see how their implementation of symmetrical encryption algorithms affected the performance depending on type of storage device used and to see how the different encryption applications dealt with the encryption of a solid state drive. I believe that I have been able to show that it is not only the type of CPU used that affects the performance when implementing encryption but also the importance of choosing the right encryption application depending on the storage device that it is intended to be used on especially when using a solid state drive where the difference is quite substantial.

6.2 Proposal for future research

The experiments performed in this thesis leaves the door open for a wide variety of additional research that can be performed such as the benchmarking of additional solid state drives in order to show if the results gathered in my thesis are generalizable or not. This is also something that should be interesting for the developers of encryption software to make sure that their application works well with newer types of storage devices such as solid state drives.

It should also be interesting to dig deeper into the cause of the problems experienced with the solid state drive in regards to the encryption implemented on it and see if this is a problem with the firmware, the particular brand, drivers or just see if the programmers implemented the algorithms and the rest of the encryption application in such a manner that it can’t fully handle the architecture of the solid state drive.

Another proposal for future research is to perform the same experiments but change a few factors such as using a CPU with built in hardware encryption or to see how RAID would affect the results. Future research could also perform the experiments in a Linux environment and include a wider range of different applications to gather a more complete performance evaluation across all encryption software applications and their various algorithms. It would also be interesting to see how the encryption of solid state drives are affected by the operating system that it is being run on and if the problems I experienced exist in Linux environments as well.

A. References

[1] S. Singh. (2000). *The code book: the science of secrecy from ancient Egypt to quantum cryptography* [E-book]. Available: http://proxy.lnu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=cat00750a&AN=lineu.529050&lang=sv&site=eds-live&scope=siteAvailable: http://proxy.lnu.se/login?url=http://site.ebrary.com/lib/linne/Doc?id=10235313

[2] J. G, Simmons, "The Data Encryption Standard and the Advanced Encryption Standard", 2012 [Online]. Available: http://www.britannica.com/EBchecked/topic/145058/cryptology/233467/The-Data-Encryption-Standard-and-the-Advanced-Encryption-Standard. [Accessed]

[3] E. Conrad, "Types of Cryptographic Attacks", [Online]. Available: http://www.giac.org/cissp-papers/57.pdf. [Accessed: 2012-04-12]

[4] S. Kramer, "Announcing development of a federal information processing standard for advanced encryption", 1997 [Online]. Available: http://csrc.nist.gov/archive/aes/pre-round1/aes_9701.txt. [Accessed: 2012-04-16]

[5] P. Bulman, "NIST Announces Encryption Standard Finalists", 1999 [Online]. Available: http://csrc.nist.gov/archive/aes/round2/AESpressrelease-990809.pdf. [Accessed: 2012-04-27]

[6] R. Snyder, "Some security alternatives for encrypting information on storage devices," in *Proceedings of the 3rd annual conference on Information security curriculum development*, Kennesaw, Georgia, 2006, pp. 79-84.

[7] M. Bauer, "Paranoid penguin: BestCrypt: cross-platform filesystem encryption," *Linux J.*, vol. 2002, p. 9, 2002.

[8] D. Elminaam Abd Salama, H. Kader Abdual Mohamed, and M. Hadhoud Mohamed, "Evaluating The Performance of Symmetric Encryption Algorithms," *International Journal of Network Security*, vol. 10, pp. 213-219, 2009.

[9] A. Nadeem and M. Y. Javed, "A Performance Comparison of Data Encryption Algorithms," in *Information and Communication Technologies, 2005. ICICT 2005. First International Conference on*, 2005, pp. 84-89.

[10] I. Corporation, "Intel® Advanced Encryption Standard Instructions (AES-NI)", 2010 [Online]. Available: http://software.intel.com/en-us/articles/intel-advanced-encryption-standard-instructions-aes-ni/. [Accessed: 2012-14-27]

[11] C. Microsoft, "32-bit and 64-bit Windows: frequently asked questions", [Online]. Available: http://windows.microsoft.com/en-us/windows7/32-bit-and-64-bit-windows-frequently-asked-questions. [Accessed: 2012-05-16]

[12] R. Richardson, "CSI Computer Crime & Security Survey", 2008 [Online]. Available:http://www.cse.msstate.edu/~cse6243/readings/CSIsurvey2008.pdf. [Accessed: April. 03, 2012]

[13] T. Goodrich, Michael and R. Tamassia, *Introduction to Computer Security*: Boston: Pearson Education, Inc, 2011.

[14] Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, 2001.

[15] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, and C. Hall, "Twofish: a 128-bit block cipher," in *Twofish: a 128-bit block cipher B2 - Twofish: a 128-bit block cipher*, ed: Counterpane systems, 1998.

[16] F. Stajano, "Nothing better than a Python to write a Serpent", [Online]. Available: http://www.cl.cam.ac.uk/~fms27/serpent/serpent-abstract.pdf. [Accessed: 2012-04-27]

[17] T. Foundation, "Introduction", [Online]. Available: http://www.truecrypt.org/docs/. [Accessed: 2012-04-16]

[18] T. Foundation, "System Encryption", [Online]. Available: http://www.truecrypt.org/docs/?s=system-encryption. [Accessed: 2012-04-16]

[19] T. Foundation, "Parallelization", [Online]. Available: http://www.truecrypt.org/docs/?s=parallelization. [Accessed: 2012-04-16]

[20] T. Foundation, "Pipelining", [Online]. Available: http://www.truecrypt.org/docs/?s=pipelining. [Accessed: 2012-04-16]

[21] T. Foundation, "Encryption Algorithms", [Online]. Available: http://www.truecrypt.org/docs/?s=encryption-algorithms. [Accessed: 2012-04-16]

[22] T. Foundation, "Cascades", [Online]. Available: http://www.truecrypt.org/docs/?s=cascades. [Accessed: 2012-04-16]

[23] J. Inc, "What is Volume Encryption", [Online]. Available: http://www.jetico.com/bcve_web_help/index.php?info=html/01_introduction/02_what_is_ve.htm. [Accessed: 2012-04-18]

[24] DiskCryptor, "DiskCryptor", May. 23, 2011 [Online]. Available: http://diskcryptor.net/wiki/Main_Page/en. [Accessed: 2012-04-18]

[25] T. Foundation, "Hardware Acceleration", [Online]. Available: http://www.truecrypt.org/docs/?s=hardware-acceleration. [Accessed: 2012-04-18]

[26] J. Inc, "New features in version 3", [Online]. Available: http://www.jetico.com/bcve3_web_help/index.php?info=html/01_introduction/04_new_in_version.htm. [Accessed: 2012-04-18]

[27] I. Corporation, "Intel® Core™ i5-760 Processor (8M Cache, 2.80 GHz)", [Online]. Available: http://ark.intel.com/products/48496/Intel-Core-i5-760-Processor-(8M-Cache-2_80-GHz). [Accessed: 2012-04-18]

[28] S. Siewart and D. Nelson, "Solid State Drive applications in storage and embedded systems," *Intel Technology Journal*, vol. 13, pp. 29-53, 2009.

[29] W. Roger, "Current Perspectives: Future hard disk drive systems," *Journal of Magnetism and Magnetic Materials*, vol. 321, pp. 555-561, 2009.

[30] Anvil, "Anvil's Storage Utilities", 2012 [Online]. Available: http://www.xtremesystems.org/forums/showthread.php?273661-Anvil-s-Storage-Utilities. [Accessed: 2012-05-04]

[31] Hiyohiyo, "CrystalDiskMark", 2012 [Online]. Available: http://crystalmark.info/software/CrystalDiskMark/index-e.html. [Accessed: 2012-05-04]