I. J. Computer Network and Information Security, 2013, 8, 19-27
Published Online June 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2013.08.03

Information Leakage Prevention Using Virtual Disk Drive

Tarek S. Sobh
Information Systems Department, Egyptian Armed Forces, Cairo, Egypt
tarekbox2000@yahoo.com

Abstract — The worst news for information technology people are computer has been stolen or lost. The actual problem is the loss of the data stored on the hard drive that can fall into the wrong hands. However, users of information system and laptops computers are facing real problems with due to intruders using attack techniques when they are connected to the network and lost or stolen computers. In order to protect your organization against information leakage you should encrypt this data by only allowing the user with access to the encryption key to view the data, authorized application usage, and control who gets access to specific types of data.

This work focuses on confidentiality of secure information storage. In addition, it presents the model to create of a Virtual Disk Drive (VDD) on MS Windows, that appear to the user (after the mounting process) as hard disks, but that are really stored as ciphered files on a file system. The proposed VDD prevents dictionary attacks and brute force attacks by incorporating a CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) in the login mechanism. The authentication method for the VDD login is based upon a 3-D image CAPTCHA. All components of this work are integrated in one security VDD tool called "SecDisk".

Index Terms — Information Leakage, Virtual Disk Drive, 3-D image CAPTCHA, Authentication, Encryption

I. Introduction

In order to protect your organization against data leakage you should fully encrypt this data by only allowing the user with access to the encryption key to view the data, authorized application usage, and control who gets access to specific types of data. In addition, you must enforce endpoint data access controls. Table 1 introduces different types of disk encryption to secure sensitive information.

A deniable file system (DFS) is one where the existence of a portion of the file system can be hidden from view [1], [2]. DFS encrypts hard drive files where files and directories are visible yet unintelligible. In a DFS, existence of certain files and directories are unavailable to the attacker [3], [4], [5], [6], [7].

Table 1: Types of Disk Encryption [17]

Full Disk encryption	File, Volume folder encryption
Sector by sector without creating temporary or backup files	Requires manual intervention to decrypt every time you want to use it
Large files will decrypt without reduced performance	Habit of creating temporary files
File encryption is normally a much slower process	Files are not safely deleted when the program exits
Full disk encryption also avoids time consuming tasks	Data can easily be accessed
Safe decommissioning of PCs	

This work introduces a security tool called "SecDisk" to find an integrated secure way to generate and hide disk portion. It contains secure important data files "bits on disks" in a Virtual Disk Drive (VDD), so as to make the maximum benefit of the security techniques that can be applied over the VDD over the hard drive. It provides an easy-to-use multiple factor of user authentication for the most sensitive encrypted data stores. This work should shift from entire disk encryption to VDD data encryption.

The paper is structured as follows: Section 2 explains some definitions about VDD. Section 3 illustrates the proposed model idea and architecture. Section 4 presents some related work and comparison with proposed system finally section 5 contains conclusion.

II Virtual Disk Drive

VDD is a temporary drive that is not physically exist on the hard disk and mounted and destroyed on demand. It is a term used when a drive is emulated as a real hard drive with a specific mount point (location in the physical disk) [8]. It is used as a physical drive from the user point of view, but treated as a file with respect to the internal system [9], [10], [11]. It has no private entry in the partition table. As shown in Fig. 1, VDD does not exist in the physical structure of the hard disk.

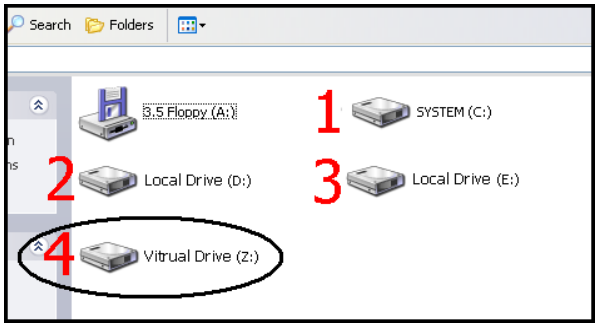

Figure 1: Partition manager snap shot

Using VDD achieves more security by mounting VDD and creating and/or moving sensitive files. When the authenticated user chooses unmounts VDD operation then VDD becomes invisible. The VDD space would be not possible to determine whether there is a VDD or not. Also, all stored files in the free space of VDD become encrypted. VDD achieves more flexibility by applying new functionality over the VDD such as applying different encryption algorithms such as IDEA, AES ...etc. Applying On-The-Fly-Encryption (OTFE) technique over stored file is important to the VDD space storage [9], [12], [13]. OTFE is an encryption technique that encrypts and decrypts files in the memory [8], [14] i.e. file resident on the virtual disk is encrypted as shown in Fig. 2.

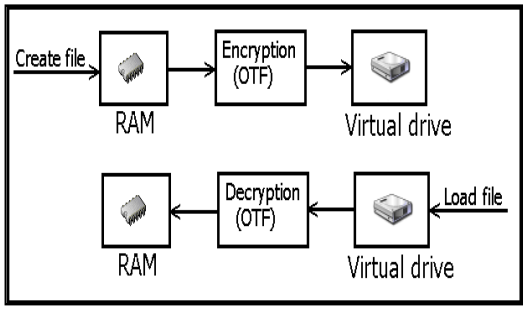

Figure 2: Secure files in VDD

III. The Proposed System Model

VDD are used due to existence of important and secret information is stored on the hard disk which may be stolen. The idea of VDD which is being introduced here is applying a special technique to create VDD inside hard drive, onto the operating system disk, on different Windows platforms.

Fig. 3 presents the proposed system model architecture. VDD model contains two main components System Driver and Interface. System driver is a system file contains the VDD and responsible to read /write operation and file encryption. Interface component is responsible to the VDD application interface, user authentication, and act as a communication layer to send commands to the driver. The mount point of the VDD is a file resident on the hard disk but completely hidden from the user, even the authenticated user, but the file still under control through the VDD service control center (SCC). SCC is an interface through which we can administrate the VDD.

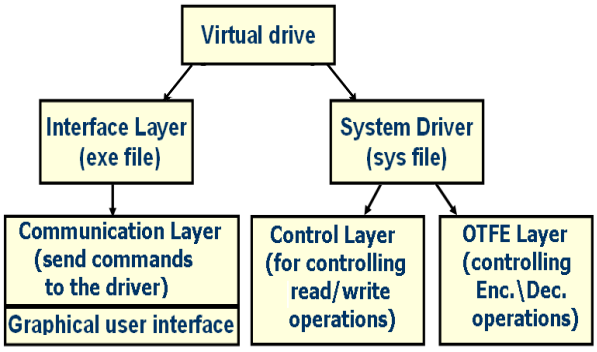

Figure 3: Proposed system model architecture

3.1 System driver

In order, to create VDD first need to make a disk area to act as container; this will be able to specify the filename and its location. It is important to set the filename container format size of disk. The space used by the VDD on the disk is encrypted for security reasons. The directory on the hard disk is uniquely identified using two pieces of information: its filename, and the path along the directory tree that you traverse to get to it, also file table (FAT) which is used to keep track of which clusters are assigned to each file.

Our main objective here is creating VDD that is visible through the windows explorer (like any other drive), but invisible as a physical drive if the hard disk explored with any commercial partition manager software such as Partition Magic. The proposed VDD creates container area, which are visible as regular disks with corresponding drive letters (for example, D:, K:, Z:, i.e. with any drive letter that is not used by another system device) and supports encryption to the all stored files in this VDD container area as shown in Fig. 4.

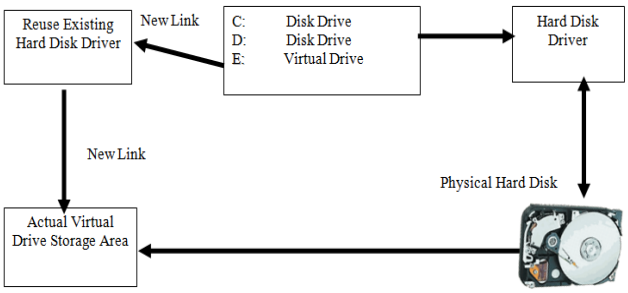

Figure 4: Windows Logical Drives Structure & VDD within IT

The proposed VDD chooses suitable storage area according to the proposed routine as shown in the flowchart of Fig. 5.

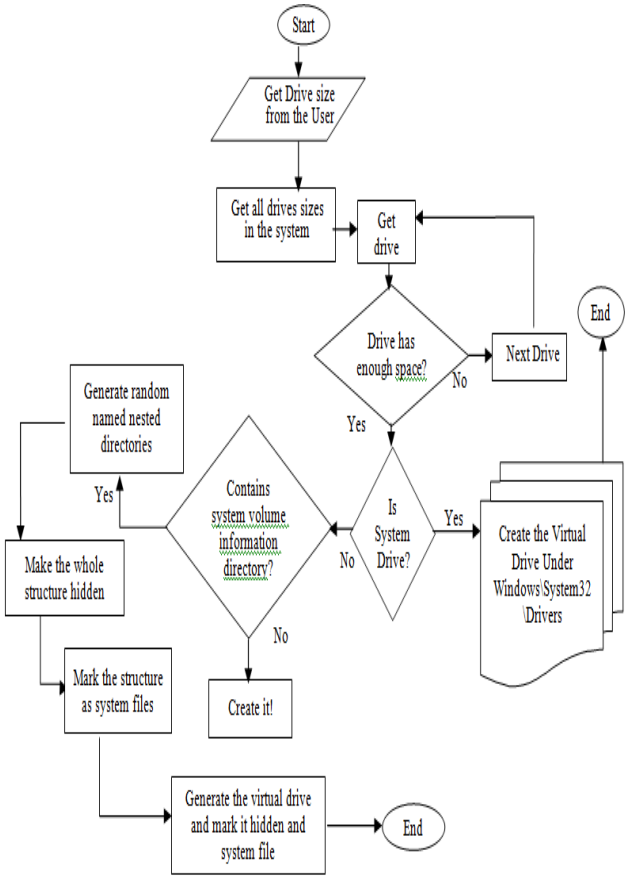

Figure 5: Choose VDD physical storage flowchart

We create and control the file and mount it as a virtual drive through building system driver. The following code represents the implementation details to create system driver of VDD:

```
        private void btnMount_Click(object sender,
    EventArgs e)
        {
            int done = 1;
            string temp = "";
            int driveSize = (int)txtVDsize.Value;

            if (rdKbyte.Checked)
              {
                driveSize =
    System.Convert.ToInt16(txtVDsize.Text);
                driveSize /= 1024;
              }
            if (rdMbyte.Checked)
              {
                driveSize =
    System.Convert.ToInt16(txtVDsize.Text);
              }
            if (rdGbyte.Checked)
              {
                driveSize =
    System.Convert.ToInt16(txtVDsize.Text);
                driveSize *= 1024;
              }
           VDN = getAvailableDrives()[1];
           if (txtVDsize.Text == "")
                MessageBox.Show("Enter virtual drive
    storage area");
           else
            {
              try
             {
                temp = GenerateVDPath(driveSize);
              }
               catch (Exception E)
              {
            done = 0;
            MessageBox.Show("there is no enough
    space on hard disk for virtual drive");
              }
               if (done == 1)
              {
              done = domount(driveSize, temp);
              if (done == 1)
                 {
                   hideVD(temp);
                   cmbVDname.Text = VDN.ToUpper();

    backgroundWorker1.RunWorkerAsync();
                 }
              }
            }
           }
```

3.2 Protect Scrambled Data Content

Types of disk encryption are file, volume folder encryption and full disk encryption (FDE). Here, to secure data the VDD we create temporary folder on the hard drive (physical drive) and save data in an encrypted way through interface layer as shown in Fig. 6. To view data in original view you must mount a VDD that demands you must authorized to mount it. All files and folders in VDD are secured through using encryption algorithms such as IDEA, AES (256) and DES. Also, all VDD contents are transparency where the VDD mount point will not be visible to the users.

Fig. 6 presents general scrambled data protection using encryption and digital signature which is used the proposed system and many other systems [8], [9], [15], [16], [17]. Scrambled content is encrypted using a key; this key is then added to the disc as a digital signature. A software loader program is added to read the digital signature, extract the key and load/decrypt the original file from scrambled content.

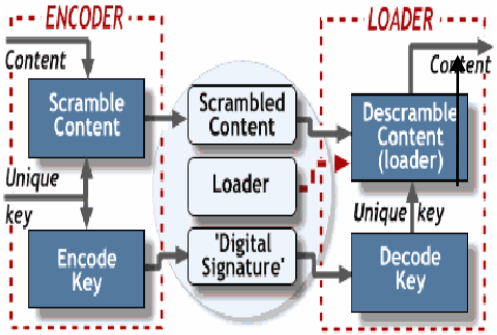

Figure 6: Scrambled data protection in VDD

In order to use files, volume folder in VDD the following steps are required:

1. Manual intervention to decrypt every time you want to use it.
2. Habit creating temporary files.
3. Files are not safely deleted when the VDD application program exits.
4. Data can easily be accessed.

It is clear from the result, shown in Fig. 7 and Fig. 8, that there is no correlation between frequency of occurrence of the plaintext and the ciphertext as proposed VDD experimentally test. As a result, it becomes difficult to predict the encryption key in differential crypto-analysis. It is confirmed that any method of brute force attack by the cryptanalyst to find out the key is highly difficult in using single letter frequency statistics to break the ciphertext without the knowledge (specific dictionary) of secret keys. This result agreed with that of [18] and others.

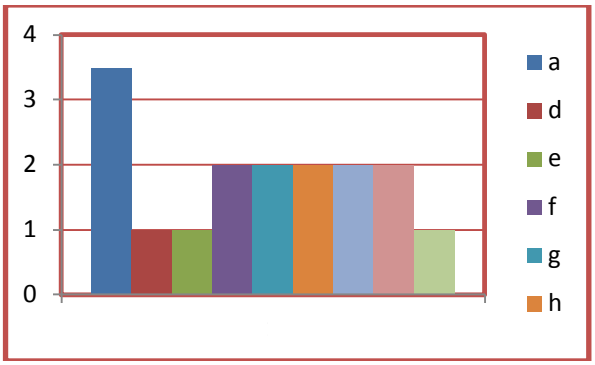

Figure 7: Characters of plaintext frequency

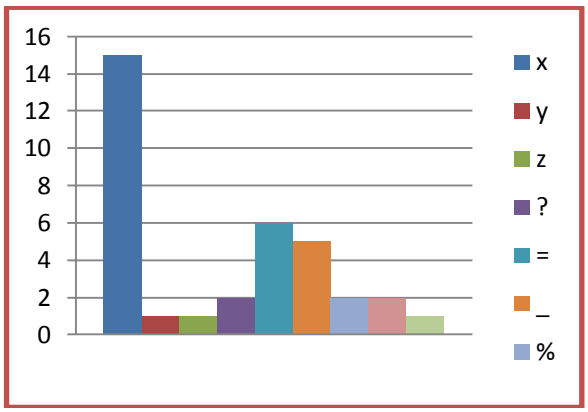

Figure 8: Characters of ciphertext frequency

3.3 Service Control Center and User Authentication

SCC is executable application for the VDD, with the following main function: 1) Authenticate the user, 2) Mount the virtual drive, and 3) Unmount the virtual drive.

In order to exploit this designed approach, the idea defines mount/unmount VDD information through CAPTCHA image authentication. Also, you need a secret key to allow the user accessing VDD content (files) in plain text (clear form).

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a popular mechanism used in Web sites to ensure humans only are interacting with the security functions of the Web site [19], [20], [21], [22]. CAPTCHA does this by producing images that include embedded numbers and letters that are not easily interpreted by automation tools.

CAPTCHA can be used in a variety of applications, among them thwarting automated responses. There are several iterations of this method. The most basic is called "Gimpy", which distorts text so that human eyes can still recognize and read the text but computers can not use automated programs to discern the letters in the text image [22].

Text-based CAPTCHAs are popular because with as little as five characters (case-insensitive letters and digits) you have 365 ~ 60 million possible combinations [19], [23], [24]. However, even after graphic distortion and degradation, some approaches have been able to ''read'' them and thus solve the test automatically around 92% of the time [21], [23]. It is possible to divide the graphic into its constituent letters. Some approaches have focused on making this division harder, typically at the expense of making it also harder to the human challenger [19], [24].

Image captchas can be more attack resistance than text captchas [25]. Common scenarios include identifying from a list of terms what best describes the displayed image. In the proposed VDD tool, one could imagine a scenario where several images are displayed and a text telling: "To select the corresponding picture which representing current VDD user account. According to Beaucamps et. al. [25] this solution has some weaknesses. First, the number of images must be big enough to resist to an exhaustive identification. Second, unless a high number of images are displayed to select corresponding image of the current user, a random login attack is likely to have a high success rate.

Detail identification technique relies on the human distinction of details inside an image [25]. So-called 3-D CAPTCHA is an implementation of such Captchas: an image is generated which assembles several features together and annotates the different parts of each feature with some text; then the user is asked to enter the text written on some part of some feature [25], [26]. If each text component is one letter long and there are enough different details to cover the alphabet and the user is asked to enter the text of several components, then there is no other way for a breaking algorithm but to identify each detail in the image. Fig. 9 is an example of such a 3-D CAPTCHA.

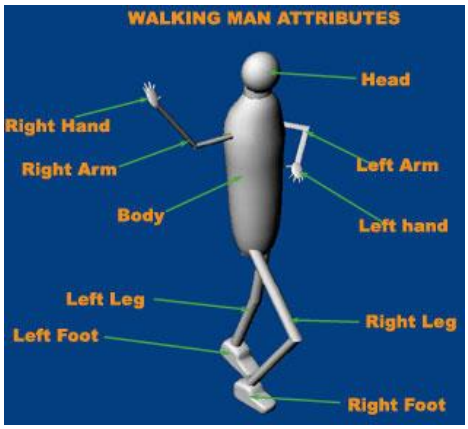

Figure 9: An example of object in 3-D CAPTCHA [26]

Michael G. Kaplan [26] introduced the following analysis to the attack against 3-D CAPTCHA:

Michael G. Kaplan [26] assumed that a malicious programmer puts in the effort to design a bot that is able

to recognize a 'flower' 30% of the time. This malicious bot will cycle through multiple 3-D CAPTCHA selecting out challenges that ask to identify a flower. The bot will correctly identify the flower and use a brute force attack to identify the remaining attributes and solve the CAPTCHA. Michael G. Kaplan [26] added this attack will be rapidly neutralized via the following mechanism: 1) for every one CAPTCHA correctly solved by this method the malicious bot will generate an enormous number of responses that correctly identify the flower but misidentify all other features. The sudden receipt of an enormous number of responses that correctly identify the flower but misidentify all other attributes will make it immediately obvious that a malicious bot is identifying the flower. 2) The compromised 'flower' is automatically removed from the library and is replaced with another unique object (e.g. an octopus, a half-peeled banana, a plate of spaghetti) taken from a reserve library of objects that have never been publicly viewed. The malicious bot will be neutralized almost instantly after starting an attack without any manual intervention on the part of the CAPTCHA operator. Finally Michael G. Kaplan [26] deduced the following: 1) a malicious bot can not succeed even if it gains the ability to recognizing the objects used to create the 3-D CAPTCHA. 2) To succeed the bot must be able to recognize every theoretical object that will instantly appear in response to an attack which is a hard task with existing computer vision technology".

Here, CAPTCHA is used as authentication program that protects VDD against bots by generating and grading tests that humans can pass but current computer programs cannot. It is applied as authentication techniques for mounting and unmounting the VDD. Also, this work provides a solution of how to integrate CAPTCHA image with VDD control center as shown in Fig. 10. The proposed CAPTCHA authentication based on one of the existing CAPTCHA image generators [27] with required modification to work as 3-D image CAPTCHA [26]. 3-D image CAPTCHA modifications are out side the scope of this paper. The proposed VDD tool uses 3-D CAPTCHA that can instantaneously detect and dynamically change to neutralize an automated attack as stated in the previous section [26].

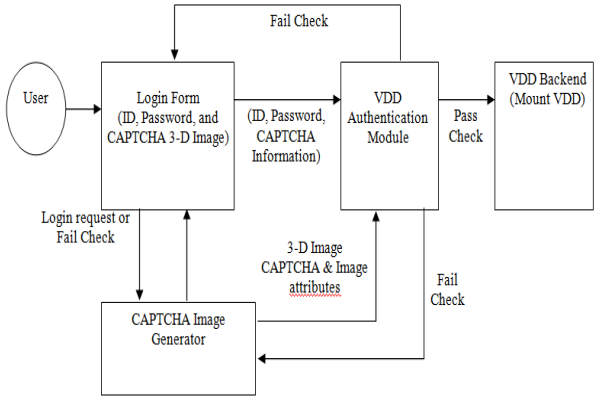

Figure 10: VDD user authentication using 3-D imag CAPTCHA

The CAPTCHA authentication process is described below.

1. The user connects to a VDD application protected by VDD authentication module. This means that the user needs to pass the authentication before accessing the VDD application.
2. The user request arrives at login form finds that the VDD application needs authentication and authorization before accessing it.
3. The login form calls the CAPTCHA image generator to generate 3-D CAPTCHA image and then send the login form. At the same time, the CAPTCHA image generator will store the same both 3-D CAPTCHA image and 3-D image attributes at the VDD authentication module.
4. The user inputs the user ID, password, and 3-D CAPTCHA image information. After the user clicks submit or login button, the login form sends the ID and password input field with 3-D CAPTCHA image information to the VDD authentication module.
5. The VDD authentication module checks to ensure the user ID, password, and the 3-D CAPTCHA image information are correct.
6. If the details are valid, the VDD authentication module allows the user to access VDD backend (i.e. mount VDD).
7. If the details are invalid, the VDD authentication module denies the user to mount VDD. It returns to login form and ask CAPTCHA image generator to generate new 3-D CAPTCHA image information.

VDD Service Control Center (SCC) is interface to windows. SCC introduces several options to users wanting to mount to exposes as driver (e.g., F:). To view data in original view you must create a VDD first that demand you must authorized to create it. To create VDD three probabilities 1) Existed virtual drive, 2) Recover previous virtual drive, and 3) Create new virtual drive. It also exposes sufficient information to the Windows operating system to allow Windows to mount and interact with the contents of VDD.

3.4 VDD System Interface

The proposed VDD is a securely managing your data. It creates a virtual space on your hard drive that appears as another drive, just like USB drives when you choose create a new VDD. Also, when you start up your computer and you want to used existing VDD choose mount an existing VDD. Finally, if you finish your task on VDD you should choose unmount VDD or when operating system shutdown your VDD unmount automatically. All VDD choices are listed in VDD interface as shown in Fig. 11a.

The proposed VDD interface is pretty straightforward and has concise instructions. There is drop-down menus choice to let VDD user chooses the nomenclature of the drive and allows the user to choose the destination folder that the aforementioned drive would point to it. Once these actions have been chosen, there is a button which

the user can click to confirm his choice. Once that is done, the drive is created (see Fig. 11b). The user can check the veracity of this by going into the explorer and the new drive is displayed. The number of partitions that a user can have is limited by the number of letters in the alphabet. One of the good things about using VDD is this software can manage data securely and safely. It is easy to install, mount, unmount or create new VDD for storage of data. These can be undone with just a click of the mouse.

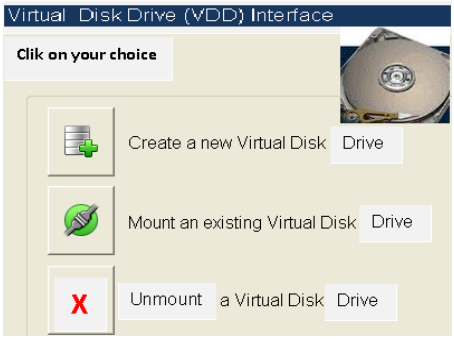

Figure 11a: VDD main page interface

Figure 11b: VDD explore new drive

You can encrypt and password protect this VDD space so that all the data stored on this is completely secure. Password protect is done by using CAPTCHA image as we stated above. Encryption is easy by using a password phrase to further secure the data on the drive. This is an excellent way to preserve and protect data from being compromised.

IV. Related Work

There are several related work focused on encrypted file systems. Some of these products are published research such as Blaze's encrypted file system for UNIX while others are commercial such as PGP Whole Disk Encryption and BitLocker, and open source systems, including TrueCrypt. Here, the basic features of such VDD are emphasized in order to indicate the relative position of the present work. Surfing the internet to find a lot of similar commercial and free products, that creates a VDD using real file (resident on the hard disk) as its mount point, also applying OTFE technique over the file such as Secure Drive, and FreeOTFE

4.1 TrueCrypt

TrueCrypt is a free disk encryption application that provides OTFE for Microsoft Windows, MacOS and Linux. It has the ability to create deniable hidden volumes [15]. These hidden volumes are optionally – hence deniably – placed inside non-hidden, regular encrypted volumes.

Outer Encrypted Volumes [15]: A regular (non-hidden) TrueCrypt encrypted volume can be stored (in encrypted form) as a file on top of a regular files system. The encrypted volume could be stored as the file C:\TCContainer. Alternately, the encrypted volume could occupy a dedicated partition on a disk. In either case, the encrypted volume is referred to as a TrueCrypt container. To decrypt this container, the user must provide the password and keyfiles that were used when creating the volume.

Hidden Volumes [15]: TrueCrypt provides a DFS through a feature known as a hidden volume. A hidden volume is a volume stored inside the container of a regular, non-hidden volume. A hidden volume requires its own password, and — if the hidden volume's password is not supplied (or supplied incorrectly) — the hidden volume's data will appear as random data. Since free space in a regular (outer) TrueCrypt volume is, according to the documentation, filled with random data, this provides plausible deniability to an attacker under the one-time access threat model.

Interface to Windows [15]: The application exposes several options to users wanting to mount a regular or hidden volume, including: mount type (whether the volume should be mounted as a fixed file system or a removable file system), write ability (read-only or not), and mount point (e.g., E:).

4.2 BestCrypt

BestCrypt is available in both Linux and Windows versions [9]. The source code needs to be compiled first. This assumes you have installed kernel header files. You will need root privileges to install the bctool, the accompanying toolset, the manpage and kernel modules using make install. This command also creates the /dev/bcrypt0 through /dev/bcrypt15 block devices. To uninstall, you can launch the ./uninstall.sh script.

Windows Encrypted Containers [19]: BestCrypt allows encrypting data with many encryption algorithms (AES, Blowfish, Twofish, CAST and others). Every algorithm is implemented with the largest possible key size defined in the algorithm's specification.

The data stored on a BestCrypt disk is stored in the container file. A container is a file, so it is possible to backup a container, move or copy it to other disk (CD-ROM or network, for instance) and continue to access your encrypted data using BestCrypt.

Any free drive letter in the system may be used to mount and to open an encrypted file-container for access. Also, you can mount file-container as a subfolder on NTFS disk. When the virtual disk is opened, you can read and write data as if it were a conventional removable disk.

There are several ways of encrypting data in version 8: Storing encrypted data in containers and accessing the data through virtual drives (as earlier versions of the software do); Encrypting set of files into a single compressed and, if needed, self-extracting. Encrypting and accessing transparently whole Windows partitions/volumes.

Linux Encrypted Containers [19]: BestCrypt will launch when you reboot, if your kernel allows you to load kernel modules. It stores the file system in a file, the so-called container, which can be mounted just like any other device, data containers can be created by nonprivileged users. Administrator, who installs BestCrypt, should be aware that users can hide some of their data from root.

BestCrypt containers not only provide more convenient access than encrypted loop devices, they offer more options. For example, bcinfo private displays the description of the private container, as entered when defining the container. bcpasswd private allows you to change the password and bctool add_passwd private adds an additional valid password, which can be deleted again using bctool del_passwd private. bcreencrypt private -a blowfish will allow you to change the encryption algorithm at a later stage. Fsck can be used to check the integrity of the container filesystem, just like a normal Ext2 filesystem, however, the bcfsck private command is required to call the tool. Also, it only protects your data until you mount the filesystem. Once mounted, users will still need to specify access privileges to prevent other accessing the encrypted filesystem. The security of encrypted filesystems is further compromised by the fact that residual data on the swap partition or the /tmp directory may allow bypassing of the encryption mechanism

4.3 Comparison

Table 2 introduces a comparative study between the proposed VDD and some existing commercial systems (TrueCrypt; BestCrypt).

Table 2: Comparative Study

Software	TrueCrypt	BestCrypt	Proposed VDD Software SecDisk
Platforms Supported	Microsoft Windows, MacOS and Linux	Linux and Microsoft Windows	Microsoft Windows Only
Encryption Algorithm	AES, Blowfish, Twofish, CAST and others	AES, Blowfish, Twofish, CAST and others	AES, IDEA, and 3DES and others
Container Decryption Needs	Password and Key files	Password, Public Key and Secret sharing Schema	Password and Key files Both are stored on e-token (Optional)
Default Windows File System	All File Systems (FAT 16, FAT 32 and NTFS)	All File Systems (FAT 16, FAT 32 and NTFS)	All File Systems (FAT 16, FAT 32 and NTFS)
Interface to Windows	Yes (Different Interfaces)	Yes (Through main window)	Yes (Through Service Control Center)
Supported Storage Devices	Hard Disks USB Flash Drive	Hard Disks, USB Flash Drive	Hard Disks Only

It is clear from Table 2 all products close in their security feature but the main difference in the container.

- Advantages of the proposed VDD:
    1. Two layers of security
        - CAPTCHA authentication (for mount, read, write…etc).
        - All file contents are encrypted using symmetric encryption algorithm.
    2. User can enlarge the virtual drive size.
    3. Encryption key is a combination of e-token data and user input password (for ex: key = F(e-token data, User input password))
    4. Virtual drive will not be visible to the users unless the user id, password and CAPTCHA authentication check are success.
    5. File (mount point) is completely hidden form the user.
- Disadvantages of the proposed VDD:

Like all other similar products, the mount point (file) of VDD is physically resident on the hard disk. Therefore, file may be deleted or manipulated by attackers if and only if the attacker found the file but this case is rare in our approach, as the file of VDD mount point is hidden from the user and if the attacker found this file of mount point, he will find all VDD content encrypted.

V. Conclusion and Future Directions

Confidentiality and security are widely regarded as prerequisites for sensitive information storage such as financial, e-government or military information. There is clearly a need to apply techniques and approaches that engender confidentiality, information integrity, security and trust for both host and end-users of the system. Other approaches such as FDE create a false sense of security, and do not provide the architectural flexibility needed to meet emerging notebooks and desktops security requirements [2].

There are different vectors of information leakage from the operating system; primary application; and nonprimary applications. The proposed VDD system protects against information leakage from the external environment that interacts with the deniable file system while the file system is mounted. This work describes applying the technique of a VDD. It provides multiple

layers of security to secure secret file on hard disk. User authentication to mount VDD and implementing encryption algorithm using a secret key to access the VDD provides first line of defense for VDD. It is easy to install and setup through wizards. It is simple client version available for initial imaging of notebooks and desktops.

The future research directions are on the integration of additional functionality of VDD with the combination of physiological biometric techniques such as fingerprints, face recognition, hand geometry, iris recognition and retina recognition, which can be used for identification or verification of identity while exchanging sensitive information for different sector areas, such as e-government.

REFERENCES

[1] A. Furche and G. Wrightson, Computer Money: A systematic overview of Electronic Payment System, pp. 456, dpunkt Verlag Fuer digital technology Gmbh, Heidelberg, FDR, 1996.

[2] C. E. Phillips, T. C. Ting, and A. Steve, "Information sharing and security in dynamic coalitions," in ACM SACMAT'02, California, USA, 2002

[3] Dhillon and j. Backhuose, "Current directions in IS security research; towards socio-organisationl perspectives," Information Systems Journal, vol. 11, pp. 127-53, 2001.

[4] D. McCullagh. Security guide to customs-proofing your laptop. http://www.news.com/8301-13578_3-9892897-38.html, 2008.

[5] Hsi, Sherry, and A. Agogino, "Scaffolding Knowledge Integration through Designing Multimedia Case Studies of Engineering Design" Engineering Education for the 21st Century: Proceedings of Frontiers in Education, FIE'95, ASEE/IEEE, pp. 4d1.1-4d1.4.

[6] J. Granick. EFF answers your questions about border searches. http://www.eff.org/deeplinks/2008/05/border-search-answers, 2008.

[7] S. Tsujii, Y. Itakura, H. Yamaguchi, A. Kitazawa, S. Saito and M. Kasahara, "Public-key Cryptographic scheme having a structure in which biological information is embedded into a secret key," IEICE Symposium (SCIS2000), D07, Jan. 2000.

[8] Faisal Nabi, "Virtual Invisible Disk Design for Information System Security", International Journal of Network Security, Vol.8, No.2, Pages.131-138, Mar. 2009

[9] D. Dong, Using SSL as an encryption tool, June 7, 2002.

[10] CREDANT Technologie, Advances in Endpoint Data Security: New Technology to Meet Security, Operations and Compliance Needs", February 2008, www.CREDANT.com

[11] Tarek S Sobh, Yasser Aly," "Effective and Extensive Virtual Private Network", Journal of Information Security (JIS), Vol.2 No.1, PP. 39-49, 2011

[12] A. Narayanan, T. Shaman, "The secure virtual computer on your keychain", Network Security, Volume 2008, Issue 7, July 2008, Pages 11-14

[13] M. Geiger and L.F. Cranor, "Scrubbing Stubborn Data: An Evaluation of Counter-Forensic Privacy Tools", Security & Privacy, IEEE Volume 4, Issue 5, Pages: 16-25, Sept.-Oct. 2006

[14] Tarek S. Sobh, "Wi-Fi Networks Security and Accessing Control", International Journal of Computer Network and Information Security (IJCNIS), Vol. 5, No. 7, PP. 9-20, 2013

[15] Alexei Czeskis, David J. St. Hilaire, Karl Koscher, Steven D. Gribble, and Tadayoshi Kohno, "Defeating Encrypted and Deniable File Systems: TrueCrypt v5.1a and the Case of the Tattling OS and Applications", http://www.truecrypt.org/

[16] Carsten Schnober, "Encrypted Virtual Filesystems with BestCrypt Locked Up Data", www.linux-magazine.com, Pages: 31-31, August 2003

[17] Joseph Belsanti, "Protecting Data-at-Rest Compliance with data and security regulations", WinMagic Inc, www.winmagic.com

[18] R.R. Sahoo and G.S. Rath, "Designing a cryptosystem by implementing reversible sequential switching M/C- a symmetric approach", International Journal of Computer and Communication Technology (IJCCT), 2(2010), 173-175.

[19] C. J. Hernandez-Castro and A. Ribagorda, "Pitfalls in CAPTCHA design and implementation: The Math CAPTCHA, a case study", computers & security Volume 29(2010), Pages: 141 – 157, 2010

[20] K. Chellapilla, K. Larson, P.Y. Simard, and M. Czerwinski, "Building segmentation based human friendly Human Interactive Proofs (HIPs)", Proceeding of The Second International Workshop on Human Interactive Proofs, 2005, pp. 1-26.

[21] M. Greg, M. Jitendra. "Recognizing objects in adversarial clutter: breaking a visual CAPTCHA". In: Conference on Computer Vision and Pattern Recognition (CVPR 03). IEEE Computer Society; 2003. p. 134–41.

[22] S Snedaker, IT Security Project Management Handbook, Chapter 10 General IT Security Plan, Pages: 261-343, June 2006, www.syngress.com

[23] J. Yan and A. El Ahmad, "Breaking Visual CAPTCHAs with Naive Pattern Recognition Algorithms", Twenty-Third Annual Computer Security Applications Conference, 2007, pp. 279-291.

[24] J. Yan and A. El Ahmad, "A low-cost attack on a Microsoft CAPTCHA", ACM Conference on Computer and Communications Security,pp.543-554, 2008.

[25] Philippe Beaucamps, Daniel Reynaud-Plantey, and Jean-Yves Marion "On the use of Internet Voting on Compromised Computers", Army Signals Academy Virology and Cryptology Laboratory, Rennes (France), March 27, 2009

[26] Michael G. Kaplan, "The 3-D CAPTCHA", http://spamfizzle.com/CAPTCHA.aspx, Browsed 2 Jun, 2010

[27] Rick Wu and Rebecca Chen, "Integrating CAPTCHA authentication technologies with WebSEAL A reference implementation by using WebSEAL EAI", 12 Feb 2008, http://www.ibm.com/developerworks/tivoli/library/t-captcha/index.html

Tarek Salah Sobh was born in Menofia, Egypt, on February 22, 1964. He received his B.Sc. degree in computer engineering from Military Technical College, Cairo, Egypt in 1987. Both M.Sc. and Ph.D. degrees from Computer and System Engineering Department, Faculty of Engineering, Al-Azhar University, Cairo, Egypt. He has managed, designed and developed several packages for business applications and security systems. He has authored/co-authored of many refereed journal/conference papers and booklet. Some of the articles are available in the ScienceDirect Top 25 hottest articles. His research of interest includes distributed systems, knowledge discovery, database system design and development, data mining, information fusion, software engineering, intelligent systems, networks and computer security, and network management.

