Chapter 2

The Context and Background

Late twentieth century witnessed a paradigm shift from analog to digital machines with the introduction of electronic transistor and the machines. As a result, the world has changed from analog to digital. The digital age can be characterized as the era which saw the application of computer technology as a tool that enhances traditional methodologies. The incorporation of computer systems as a tool into private, commercial, educational, governmental, and other facets of modern life has improved the productivity and efficiency of modern life (Reith et al, 2002). The digital revolution has paved the way for a new technology called Information Technology (IT), and industry, government and indeed society are becoming critically dependent on Information Technology (Benjamin et al, 1998). Although the computer reigns supreme in the digital domain, it is not the only digital device. An entire constellation of audio, video, communications, medical and photographic devices have also been made digital, to work closely associated with the computer (FBI, 2000).

2.1 Cyber crime

As a knock-on effect or unpleasant sequel of the digital revolution, digital equipment too is vulnerable to widespread abuse, and this has given rise to criminal charges and suits against digital or cyber criminals. A wide spectrum of abuses of digital technology products exist. For example, spam is the abuse of email system, software piracy is the abuse of software system, file slacking is the abuse of operating system, IP spoofing is the abuse of networking technology, DoS (Denial of Service) attack is the abuse of the Internet, and pornography is the abuse of digital video technology, just to name a few. While all sort of digital equipment are generally favourite tools of criminals, a computer probably can be their most favourite tool because of many reasons such as the general
proliferation of computer use, the computer’s power to encrypt, keep and deliver data anywhere in the world and the computer’s ability to interface itself (for data transfer) with the entire constellation of digital (audio, video, communications, medical and photographic) devices.

One type or example or form of cyber crime is computer abuse. While bashing up one’s own PC in frustration is an abuse of digital equipment not amounting to a cyber crime, any abuse of digital equipment, if done with mens rea resulting in public harm, can be treated as a cyber crime. Other forms of computer criminality include the copyright infringement through software piracy, Internet fraud and marketing scams, identity theft, the creation and transmission of child pornography, and the compromising of network security. One of the oldest definitions of computer abuse may be that of Donn Parker and his definition of computer abuse is dated back to 1976, an era of pre-World Wide Web and low bandwidth. Parker, according to Anthony Reyes (2007), is usually cited for favouring the term computer abuse and describing computer abuse as “…any incident involving an intentional act where a victim suffered or could have suffered a loss, and a perpetrator made or could have made a gain and is associated with computers” (Parker, 1976, cited in Reyes 2007, p.24).

Cyber crimes are generally malicious activities by criminals seeking to (intentionally or sometimes unintentionally) exploit or disrupt an Information Technology (IT) system, for mischief, financial gain or more sinister motives (Benjamin et al, 1998). In principle, cyber crime encompasses any crime capable of being committed in an electronic environment (UNCPCTO, 2000).

Cyber crimes are not necessarily new crimes, but rather traditional crimes committed by exploiting the power, user-friendliness and availability of digital equipment as well as accessibility to information. Computer crimes can originate when unethical hands attain user proficiency of any digital equipment and thus, gain easy access to digital information (Reith et al, 2002). Thus, some of the cyber crimes are traditional crimes that involve a computer (or any digital equipment) just as a tool for committing traditional crimes, thanks to the user-friendliness and easy availability of modern digital equipment. For example, (physical) rape is a traditional crime but it can use the digital media as a means
of perpetration. In such cases, the rape suspect’s email communications (digital evidence captured by the e-mail server) or mobile phone call / message details (again, digital evidence captured by the mobile service provider’s computer server), are often used as valuable evidence to establish culpability.

Cyber criminals often are mercenaries of cyberspace, motivated mainly by money and rarely by ideology or politics (Stephenson, 2000, p83). Many are also accomplished criminals in non-computer areas, such as fraud, sabotage, and industrial espionage. The computer simply adds a new dimension to their acts.

### 2.2 Categorization of cyber crimes

Computers can assume a variety of roles in the commitment of a crime and each of these roles can raise novel investigative and prosecution-related issues because of the unique attributes of computers and of the electronic evidence they hold (Marcella et al, 2008). Based on these roles, there are perhaps several ways of looking at and categorizing cyber crimes (computer crimes), and for this, one could use different taxonomic approaches too like the nature of the crime, its intent, tool that it uses or the analyst’s purpose of classification. Cyber criminal activities could also be looked at in terms of their intentionality (for intellectual gratification, for destructive purposes etc.) or the technology involved (network crimes, software crimes etc.). A computer can be (1) the subject of a crime (by being stolen or damaged); (2) the site of a crime (such as when a child is solicited for sex in a chat room) or (3) the instrument of a crime (such as when it is used to store information illegally) (Hinduja, 2003, p1). The categories may or need not be exclusive and the criminal activities can overlap. For instance, spreading virus, even though destructive, has been found to provide (perverse) gratification for the criminal.

Donn Parker (Parker, 1976, cited in Reyes 2007, p.24) is considered to be the person who presented the first definitional categories for computer crime. Donn Parker’s definitional categories of computer crime (based on four different roles of the computer) are thus: (1) The computer is the object, or the data in the computer are the objects, of the act, (2) The computer creates a unique environment or unique form of assets, (3) The computer is the
instrument or the tool of the act and (4) The computer represents a symbol used for intimidation or deception (Parker, 1976 cited in Reyes, 2007, p.25) This categorisation of pre-World Wide Web and low-bandwidth era seems to be broad enough to encompass modern computer crime as well. However, as noted by Eoghan Casey in his book ‘Digital Evidence and Computer Crime’ (Casey, 2004), Parker’s definitional categories of computer crime do not uphold the role of computers as a source and/or storehouse of digital evidence.

The US Department of Justice, upholding the role of computer as a source and/or storehouse of digital evidence, categorizes computer crime in three ways:\(^9\): (1) Using computer as a target: (for example, spreading viruses) (2) Using computer as a tool to commit "traditional crime" (for example, credit card fraud) and (3) Using computer as an accessory (for example, to store illegal or stolen information). This classification is based on the role of computer as the instrument of crime.

A computer crime that involves cyber space or involves the use of cyber space can be called a cyber crime. The CRS report (CRS, 2006) for US Congress on “Cybercrime, The Council of Europe Convention” (prepared by Kristin Archick, a US government specialist in European Affairs), mentions fours categories of computer-related (cyber) crimes: (1) fraud and forgery, (2) child pornography, (3) copyright infringements and (4) security breaches (such as hacking, illegal data interception, and system interferences that compromise network integrity and availability). While all the above four categories of crimes involve computer as the medium to conduct crimes, the fourth category additionally involves computers as the instrument to pose threats to computer technology itself.

The Tenth United Nations Congress on the Prevention of Crime and the Treatment of Offenders (UNCPCTO, 2000) has categorized five offenses as cyber-crime: (1) Unauthorized access to confidential or private information, (2) Damage to computer data or programs, (3) Acts of sabotage to hinder the functioning of a computer system or network, (4) Unauthorized interception of data to, from and within a system or network and (5) Computer espionage.

Like all crimes, not all cyber crimes need to be of equal magnitude or culpability and some are inherently and/or potentially more harmful than others. The UN Congress (UNCPCTO, 2000), for instance, emphasizes the following categories of computer crime: (1) Financial (crimes that disrupt company’s ability to conduct e-business), (2) Piracy (crimes of copying copyrighted material without an explicit permission), (3) Hacking (crimes of gaining unauthorized access to a computer system or network and in some cases making unauthorized use of this access), (4) Cyber-terrorism (a type of hacking, designed to cause terror, violence against persons or property, or at least cause enough harm to generate fear), (5) Online Pornography (possessing or distributing child pornography against law and distributing pornography of any form to a minor) (UNCPCTO, 2000).

Robert Taylor et al (2006), present four categories of cyber crime and they are (1) The computer as a target (for example, a Denial-of-Service attack), (2) The computer as an instrument of the crime (for example, stealing personal information stored digitally), (3) The computer as incidental to a crime (for example, trading of child pornographic materials) and (4) Crimes associated with the prevalence of computers (for example, software piracy). This categorization focuses on the (use of computer) technology.

David Wall (2001) categorizes cyber crimes into four established legal categories as (1) Cyber-trespass (for example, hacking, virus), (2) Cyber-deceptions and thefts (for example, credit card fraud, software piracy), (3) Cyber-pornography and (4) Cyber-violence (for example, cyber stalking, spreading anti-national leaflets). Majid Yar (2006) explains that David Wall’s classification can be seen to sub-divide cyber crime according to the object or target of the offence: the first two categories comprise ‘crime against property’, the third covers ‘crime against morality’ and the fourth relates to ‘crime against the person/country’.

Thus, through various categorizations mentioned above, ideas, intents and instruments behind cyber crimes are properly recognized, differentiated and understood and such categorizations can be beneficial for better research, inference and decision making concerning various technological abuses of digital equipment.
2.3 Digital evidence in cyber crime

Abuse of digital equipment leads to cyber crimes and leaves behind digital evidence, which is the data or information of probative value stored or transmitted in digital form (FBI, 2000). As data or information lies in the memory of the digital equipment, so does digital evidence. In the digital world, people leave digital footprints of their activities and so digital equipment can be a source of evidence to infer / establish their actions and intentions. As more and more transactions from the commercial world as well as government and private individuals exist only in digital form, the only way in which one can prove that something (criminal) has happened (or failed to happen) is via digital evidence (IAAC, 2009). Moreover, while any digital equipment generally can be treasuries or coffers of digital evidence, computers in particular are popular for their power to keep criminal data in encrypted form (criminal-friendly but investigator-unfriendly) and if such evidence can be obtained and decrypted, it can be of great value to criminal investigators.

Any digital data that helps the investigation of a case is digital evidence. Ueli Maurer (2004) has described digital evidence as a ‘bitstring’ which can serve as evidence in a certain context. According to him, there are two types of digital evidence: (1) Digital recordings (for example, a digital image, video, or sound recording, which are projections of physical reality) that are normally evaluated and interpreted by human beings after converting them back to some kind of physical form (e.g., a digital image into an image on a computer screen) and (2) Digital evidence strings (for example, a digital signature), which are verified by evaluating a well-defined (and unambiguous) mathematical function (for example, the signature verification function relative to some public key). Digital evidence strings may perhaps appear to be more useful than digital recordings because they can be checked automatically and are, therefore, unambiguous. However, the main usefulness of digital recordings, even if they are easy or only moderately difficult to forge, is that they are human-interpretable as a physical reality and can, hence, meaningfully be confirmed or denied by a person, if necessary, under oath (Maurer, 2004).
Yasinsac et al appreciate the presence of digital data (in computer forensics) in both its dynamic (for example, network traffic) and static (for example, a digital video clipping) states and both these types of data can be potential digital evidence (Yasinsac et al, 2003). While the dynamic digital evidence is often relevant to intrusions, static digital evidence helps to uncover the crime’s conventional aspects. (Yasinsac et al, 2003). In short, right from a computer network down to a thief’s abandoned digital diary, any digital equipment is rich in digital evidence in many forms for a wide range of computer crimes or misuse, including but not limited to theft of trade secrets, theft of or destruction of intellectual property, fraud, child pornography, disputes of ownership, prevention of destruction of evidence etc. (Williams, 2005). Some classic examples of these forms are: the left-behind data to trace email to senders, history of website visits (both dynamic) and digital images, data on shopping habits, digital address book (all three static).

A digital data (digital object) is a set of sequences of bits; it is accompanied by a variety of metadata related to that object, and with proper storage management, replication, and refreshing, this set of sequences of bits can be maintained for a long period of time (Lynch, 1999). Properly maintained digital data can be valuable potential digital evidence for the cyber forensic expert. The chances of success in cyber litigation or successful cyber criminal prosecution by law enforcement agencies depend heavily on the availability of strong digital evidence (IAAC, 2009).

US courts define all digital evidence as physical evidence, which means that digital data is a tangible object, like a weapon, document, or visible injury, that is related to a criminal or civil incident (Nelson et al, 2004).

2.4 Preservation of digital evidence

The ultimate usefulness of digital evidence depends mostly on the degree of care that their custodian takes, for preserving the digital evidence. The purpose of such preservation is to protect data / information of enduring value for access by present and future generations (Conway, 1990: 206). Since every piece of data can potentially be evidence in the event of a dispute or crime or court case, the preservation of data is very important in legal field.
Digital evidence is often highly volatile and is easily compromised by poor handling. As
digital evidence (such as digital signature) is of crucial importance in the emerging
digitally operating economy because of its easiness to transmit, archive, search, and
verify, there is scope for a systematic treatment of digital evidence in any organization
(Maurer, 2004). While many sensible organisations have arrangements in the event of
fire, flood, failure of electricity and telecommunications services or acts of terrorism,
very few have thought-through plans to identify, collect and preserve digital evidence in
forms which will prove robust against testing in legal proceedings (IAAC, 2009). For
instance, during 2007 and 2008, a number of UK government departments suffered
catastrophically by way of losses of computers, memory sticks, and CDs. Losses of
computers, memory stick and CDs can turn out to be instances of loss of digital data and
thus, they can be instances of loss of potential digital evidence of a possible cyber crime.
These losses of UK data were followed by a series of official reports in UK. The reports
from Poynter (on the loss of 25 million HMRC records on two CDs\textsuperscript{10}), Burton (on loss of
personal data in the UK Ministry of Defence\textsuperscript{11}), Thomas/Walport (on data sharing\textsuperscript{12}) and
Hannigan (on data handling procedures in Government\textsuperscript{13}) are instances of such official
reports. All these reports point to the need of preserving highly volatile digital evidence,
procedurally, with policies for handling and sharing such data.

The very nature of digital devices makes them inherently fragile (Kornblum, 2002) and
so, this can be a reason to doubt the ‘integrity’ and ‘fidelity’ of digital evidence. While
integrity answers the question whether the structure of the data remained unchanged since
it was first obtained, fidelity answers the question of how closely the data accurately or
truthfully represents fact or factual events (DFRWS, 2001) and the fidelity generally
depends on the integrity.

\textsuperscript{10} \url{http://www.hm-treasury.gov.uk/media/0/1/poynter_review250608.pdf}, visited on 7th
April, 2010
\textsuperscript{11} \url{http://www.mod.uk/DefenceInternet/AboutDefence/CorporatePublications/PolicyStrategy
andPlanning/ReportIntoTheLossOfModPersonalData.htm}, visited on 7th April, 2010
\textsuperscript{12} \url{http://www.justice.gov.uk/docs/data-sharing-review.pdf} visited on 7th April, 2010
\textsuperscript{13} \url{http://www.cabinetoffice.gov.uk/media/65934/data_handling.pdf}, visited on 7th April,
2010
2.5 Integrity of digital evidence

Integrity of digital evidence is generally a major concern of cyber crime investigators because digital evidence can be erased or changed, sometimes even without a trace, creating more hurdles for the investigator and frustrating the investigator's efforts to get at the facts. Many tools and methods exist that allow almost anyone to modify almost any attribute associated with digital data (DFRWS, 2001) which can be one reason to cast doubt on (or at least occasionally suspect) the integrity of digital evidence. This can be explained with an example from the field of digital imaging. Camera and software companies have created products to provide various sorts of archive images, audit trails, and image authentication systems. Some of these products have provided the basis for the present raw files and audit trails. Most camera manufacturers now offer a raw file format in digital cameras. The benefit of raw formats, as related to digital image integrity, is that they are virtually unalterable (Reis, 2006). Raw file formats are ‘read-only’, which makes them difficult to alter without leaving traces that experts can detect. With the DNG format, one can now take that raw file and embed it (along with any adjustments made) in a raw file processor, and archive this as a single file. The DNG file format provides an open source format that meets the needs of the forensics, medical, military, and industrial fields for archiving. The Adobe technical paper\textsuperscript{14} predicts that as more software and hardware manufacturers support the DNG format, this format can become the standard for archiving raw files in a secure manner that will meet the needs in fields in which image integrity is essential (Reis, 2006).

Integrity of digital evidence has been a prime topic of discussion among forensic researchers in the last decade. The first digital forensics research workshop held in Utica, New York, emphasizes that accepted, standardized procedures are essential to maintain integrity and foster fidelity in digital evidence collection and that adequate training (based upon these standards) is essential to develop the interpretive skills of the investigation experts to observe the syntax, semantics, and pragmatics of the data in order

\textsuperscript{14} \url{http://www.adobe.com/digitalimag/pdfs/phscs2ip_digintegr.pdf}, visited on 24\textsuperscript{th} May, 2010
to yield superior results (DFRWS, 2001). Since 2001, DFRWS continues to be held yearly to discuss matters related to creating standards in digital data integrity\textsuperscript{15}.

In short, any crime involving digital equipment leaves digital evidence and success of proving any such crime largely depends on the preservation of the digital evidence with high degree of integrity.

2.6 Cyber laws and digital evidence

In order to bring cyber criminals into the court of law, almost all countries have formulated cyber laws, by taking cue from the 1996 model law on electronic commerce and electronic communication adopted by the United Nations Commission on International Trade Law (UNCITRAL). These cyber laws are also consequences or materialization of the recommendations\textsuperscript{16} of the general assembly of United Nations that all states should give favourable considerations to the said model law when they enact or revise their laws. As a result\textsuperscript{17}, in India, the Indian Information Technology Act-2000 (Indian IT Act, 2000) was born. This act primarily provides legal recognition for transactions carried out by means of electronic data exchange, and other means of electronic communications, commonly referred to as “electronic commerce”, which involve the use of alternatives to paper-based methods of communication and storage of information. Secondly, the IT Act 2000 facilitates electronic filing of documents with Government agencies and finally, it amends the Indian Penal Code, the Indian Evidence Act (1872), the Bankers’ Books Evidence Act (1891) and the Reserve Bank of India Act (1934) and for matters connected therewith or incidental thereto. In short, Indian IT Act 2000 basically authenticates only e-commerce (including electronic stock trading) and digital signature. The IT Act 2000 was amended in 2008 to include more sections to cover other cyber crimes. For instance, Section 66 of IT Act 2000 has been expanded to

\textsuperscript{15} http://www.dfrws.org/, visited on 24\textsuperscript{th} May, 2010
\textsuperscript{16} See UN general assembly resolution number A/RES/51/162. dated 30\textsuperscript{th} January, 1997
\textsuperscript{17} The Indian Information Technology act (2000, p.1) says “....WHEREAS the General Assembly of the United Nations by resolution A/RES/51/162, dated the 30th January, 1997 has adopted the Model Law on Electronic Commerce adopted by the United Nations Commission on International Trade Law;”
include sections 66A (offensive messages), 66B (theft cases involving stolen computer, laptop, mobile etc), 66C (Identity theft), 66D (Impersonation), 66E (Voyeurism) and 66F (Cyber Terrorism), and section 67 has been expanded to include Sections 67A (Sexually explicit content), 67B (Child Pornography), Section 67C (security monitoring of Cyber Cafes). Further, the police is more equipped with section 69A (blocking objectionable websites) and 69B (monitor cyber transactions). Section 72A of IT Act protects data handled under a contractual arrangement by companies (Indian IT Act, 2008). In India, most cyber criminals are often booked under various sections of Information Technology Act-2000 and Indian Penal Code.

2.7 Software Piracy as cyber crime

One instance of a cyber crime is the copyright infringement of software. The copyright infringement of software (often referred to as software piracy) amounts to several practices which involve the unauthorized copying of computer software\(^{18}\). The term software piracy usually refers to some form of unauthorized copying of software (Law & Wong, 2005 cited in Holsapple, 2008). It is an illegal act of copying software for any reason, other than back up, without explicit from and compensation to the copyright holder (Gopal et al, 1998). Software falls under the purview of intellectual property and the copyright law generally prohibits illegal duplication of intellectual property in most countries. The pervasiveness of the illegal copying of software is a worldwide phenomenon (Poddar, 2004) and it is generally considered a (cyber) crime against global economy. Software piracy is extensive probably because it is simple, economical and profitable to copy software programs. It not only has a profound effect on the users of the software, but also on the software industry as a whole, and has a tremendous impact on the development of digital intellectual property and technology. The act of piracy not only disregards the intellectual effort and right of possessions of the author but also deprives the author of the software of a fair return for their work (Thomas et al, 1990).

According to Poddar (2004, p1), the two types of piracy that are widely observed are (i) end user piracy (ii) commercial / retail piracy. In end user piracy, pirates are the end users and piracy is performed usually with the objective of using the pirated version, free of cost. In commercial/retail piracy, pirates sell the pirated products to consumers with a profit motive, which is a commercial exploitation of the original product. Poddar further observes that end user piracy is prevalent more or less everywhere, whereas commercial piracy is more prevalent in the poor and developing countries where the laws against piracy or in general enforcement against copyright violations are rather weak. In a study using country-specific statistics, Gopal and Sanders (2000 as cited in Holsapple et al., 2008) conclude that countries with lower per-capita gross domestic product (GDP) have higher rates of piracy as compared to those with higher per capita GDP, and lack of price discrimination on the part of software vendors is one of the contributing factors in spreading piracy.

Software piracy is generally done through the online transmission mediums like the World Wide Web, the USENET newsgroups, instant messaging programs, chat programs, and FTP programs (which are used both to serve and to retrieve software). Software piracy occurs in many other ways (through other transmission mediums) also, including unauthorized burning of CDs, casual sharing of software among friends, or over networks. The rapidly falling cost of peripherals such as CDs, floppy discs, and thumb drives (memory sticks) has made this problem all the more prevalent (Hinduja, 2001; Andres, 2006). A computer user may use a combination of these ways and means for piracy. For instance, Sameer Hinduja (2003) observes that roughly 1 to 3 (exactly 1.05 to 2.83) of various transmission mediums (ways and means) are used to pirate by university students in US.

One of the major contributing factors for the rapid spread of piracy is the easy access to Internet bandwidth (Peace et al., 2003). While the almost prehistoric 300-bps modem would take over 3 years to transfer one CD (700 MB) of data over the Internet, and 9600-baud modems would take a little over a month, a 56.6-kbps model would be able to do that in a little under a week. The advent of broadband, however, has made this transfer possible in a matter of a few hours. It comes as no surprise that pirates relish broadband (Craig et al, 2005).
The level of piracy across various markets varies a great deal. In some markets, piracy is rampant while in some other markets, piracy is rare\textsuperscript{19}. There exist some empirical studies (see Gopal and Sanders (1998), Husted (2000), and Holm (2003)) to explain the varying piracy rates across countries and regions. Sougata Poddar of National University of Singapore has attempted to develop an economic model to explain the varying rates of piracy across nations, and more specifically, between the developed and developing nations (Poddar, 2004).

Software piracy has apparently become a major drain on the global economy (Limayem, 2004). According to the Business Software Alliance (BSA, 2007 as cited in Holsapple, 2008), the estimated world piracy rate for business software applications alone was 35% in 2002, leading to losses of nearly $40 billion. According to another study by Business Software Alliance (BSA, 2004), while $59 billion in commercially packaged software was sold in the global marketplace in 2004, $90 billion was actually installed and the remaining $31 billion can be attributed to pirated software. Furthermore, the same study reveals that for every two dollars worth of software that is legally sold, one dollar worth is purchased illegally. According to a very recent study by BSA (BSA, 2009), in 2008, the rate of PC software piracy dropped in about half (52 percent) of the 110 countries studied and remained the same in about a third (35 percent) but, the worldwide piracy rate went up from 38 percent in 2007 to 41 percent in 2008. This BSA study further found that the global rate rose for the second year in a row because PC shipments grew fastest in high-piracy rate countries. According to another BSA report (BSA press report, 2009), from the approximately 45-50 end-user civil actions initiated jointly by its member companies (from January 2008 onwards) against companies and organizations using pirated software in India, the approximate value of pirated software found in these companies could be valued at Indian Rupees 85,78,30,000. While it has been suggested that some of these figures may, in fact, be exaggerated (see, for example, Hayes, 2006 cited in Holsapple, 2008), the central argument still holds: software piracy affects global economy and is a big enough problem to be taken seriously.

\textsuperscript{19} Piracy rates defined as the ratio of the number of pirated copies to total installed copies, vary from 24 percent in US to 95 percent in Vietnam in the year 2002. (Source: IPR report for BSA 2003 as cited in Poddar, 2004)
2.8 Software copyright laws

Copyright regulations give the most basic protection from piracy for computer software (Friedman et al., 1997). Under copyright laws of most countries, software is generally considered as a "literary work" and is, thus given all of the protections of literary works, such as novels or poems. Additionally, under international agreements such as WTO’s TRIPS, any software written is covered by copyright laws because computer programs are classified as literary works (ESALab, 2007). According to these laws (that varies from country to country) and agreements, the owner of the copyright of software often has the exclusive right to make and distribute copies, and to create derivative works. (The owner of the copyright is generally the author of the software. If an employee creates a work as part of his employment, the employer is often considered to be the "author" for copyright purposes.) The author acquires a copyright on a work as soon as it is fixed in a tangible medium of expression, which, in the case of software, means as soon as it is written down on paper or electronically on the computer in some readable form (Friedman et al., 1997). Copyright generally lasts for the life of the author plus a few years\(^{20}\) if the author is a natural person, or for a long time (say, 150 or 170 years) if the author is a corporation; but, again, these legislations vary from country to country. The right to create derivative works of the copyrighted software gives the owner of the copyright, the right to create updates, and new versions of the copyrighted work. The right to make and distribute copies of the copyrighted work can be important, since those rights are the strongest weapons against pirates.

While the Indian Copyright Act (1957) and the Indian Information Technology Act (2008) are yet\(^{21}\) to be amended to include sections to curb intellectual property protection of computer programs, the United States Copyright law is often quoted as an example of good legislation for intellectual property protection available to computer programs. In the United States copyright law, according to Christina M. Reger (2004, p.221),

\(^{20}\) In India it is “Life of the author + 70 years”, according to the 1999 amendment of the Indian Copyright Act 1957. In the US also, it is generally “Life of the author + 70 years”, with exceptions based on date of publishing.

\(^{21}\) As on 29th May, 2010
“Currently, there are three prominent methods available for protecting the various components of computer software: copyright, patent, and trade secret. Copyright law protects the literary work (the computer code); patent law safeguards the utilitarian aspects; and trade secret law theoretically protects the idea”.

Software is a machine whose medium of construction happens to be text. This captures the inseparably dual nature of software: it is inherently both functional and literary, both utilitarian and creative. This seems a paradox only in the context of a conceptual framework of copyright laws (e.g. US Laws) that cannot conceive of such objects (Davis et al, 1996, p21).

Even though copyright protection is available for most parts of the software, software interface (an important part of the software) is often an exception. Generally, software interfaces were either published (free of IP protection) or were maintained as trade secrets by commercial distribution of programs in unreadable binary code. According to Samuelson (2008, p1), in the US, “….after courts rejected copyright protection for interfaces, a patent protection was sought for interfaces…… No other intellectual product has traversed as many forms of IP protection as software interfaces and none have transformed the law so much as it passed through these forms”.

US attorney Joseph Francis Agnelli III (2008, p.1) says “United States Copyright law emerged as the most applicable intellectual property protection available to computer programs. This was due to the fact that computer programs and some of their related components tend to fit rather well into (US) copyright law. For example, the computer programs and their user manuals have been considered writings, databases have sometimes been considered copyrightable compilations, and the visual displays on computer monitors can be considered graphic works. Also, the source code of a computer program, or the collection of statements or declarations which allows the computer programmer to communicate with the computer using a reserved number of instructions, and the object code, or the representation of code that a compiler or assembler generates by processing a source code file, are both copyrightable subject matter”.

On the European Community initiative in protecting intellectual property in software, Agnelli (2008, p.9) observes thus: “Research conducted on this issue, and how it has been
approached by the European Community, revealed a very impressive approach as to how the European Community (EC) handled the complexity of the situation that is computer programs. On May 14, 1991, the EC’s Council of Ministers adopted the Directive on the Legal Protection of Computer Programs. With the enactment of this Software Directive, the EC leapt ahead of the United States and Japan, two world leaders in computer programming and technology……. With the passing of the Software Directive, the EC has essentially created a sui generis body of law, which is the approach that this writer suggests the United States Congress must take. Several Articles of the directive mirror United States copyright law. Specifically, Article 4(a) of the Directive includes some very special language which the United States must implement into its own sui generis legislation”.

Ironically, some owners of the copyrighted software feel that illegal copying of their copyrighted software by pirates is a "lesser evil" than the pirates actually buying (or illegally copying) a competitor's software. For instance, Jeff Raikes, a Microsoft executive, states that "If they're going to pirate somebody, we want it to be us rather than somebody else……. We understand that in the long run the fundamental asset is the installed base of people who are using our products. What you hope to do over time is convert them to licensing the software". Here, although software piracy is treated by the owner as a criminal offence against him, he uses the piracy as a potential opportunity to sell a legal version of the software to same pirate. To convert a pirate into legal licensee of the pirated software, these companies use (their own sponsored) organizations. Business Software Alliance (BSA) is one such organization which “acts to protect software providers’ intellectual property rights, enforce software copyright legislation, and encourage compliance” by tackling end-user copyright infringement in the workplace where software has been installed, but the required license has not been purchased. In short, enhancing own market base by first encouraging piracy of the copyrighted product and later forcing the pirates to buy required licenses (of this copyrighted product), can be seen as one of the successful marketing strategies in software industry.

23 http://www.bsa.org/country/BSA%20and%20Members.aspx visited on 29th April, 2010
2.9 Digital forensics

Computer crimes are on the rise and unfortunately only less than two percent of the reported cases result in conviction (Baryamureeba, 2004). Convictions often crucially depend on the appropriate conversion of digital evidence into a legally acceptable and convincing form. The legal acceptance of the outcome of the computer crime investigation generally depends on the selection, adoption, and usage of appropriate process (methodology and approach) one adopts in the investigation. Incomplete or inconclusive results may lead to wrong interpretations and conclusions which in turn may finally help a computer crime culprit walk scot-free or force an innocent suspect suffer negative consequences. Thus, forensics of digital equipment (involved in the crime) has great significance.

Forensic science (often shortened to forensics) is the application of a broad spectrum of sciences to answer questions of interest to a legal system. The American Heritage Dictionary of the English Language 2000, (as cited by Hall, 2005) defines forensics as "the use of science and technology to investigate and establish facts in criminal or civil courts of law". Generally, forensics encompasses the accepted scholarly or scientific methodology and norms under which the facts regarding an event, or an artifact, or some other physical item (such as a corpse) are ascertained. According to Erbacher et al, the term computer forensics has many synonyms and contexts. While the synonyms like Cyber forensics, Forensics, Security Forensics, Digital Forensics, and Forensic Analysis are very common, the ones like Media Analysis ( old definition, referring to examining stand alone computer hard disks for digital evidence) and network forensics (which is the capture and analysis of computer network traffic to unearth digital evidence) also exist. (Yasinsac et al, 2003).

Forensics of digital equipment is digital forensics (Reith et al, 2002). Digital Forensics can be and indeed has been defined in different words by different people and not all definitions are by any means complete or comprehensive. More formally, it is “the gathering and analysis of digital information in an authentic, accurate and complete form.

---

for presentation as evidence in a civil proceeding or a court of law” (Williams, 2005). Dorothy A Lunn defines digital forensics as "The employment of a set of predefined procedures to thoroughly examine a computer system using software and tools to extract and preserve evidence of criminal activity." Digital Forensic Research Workshop held in Utica, New York, formally defined digital forensics as the use of scientifically derived and proven methods towards the preservation, collection, validation, identification, analysis, interpretation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal or helping to anticipate the unauthorized actions shown to be disruptive to planned operations (DFRWS, 2001).

In his introductory paper on cyber forensics, Gregory A. Hall (2005, p.16) makes available the following 7 definitions/introductory notes for digital / computer / cyber forensics

i. “Computer forensics is the analysis of hard drives, CDs, floppy disks, Zip and Jazz disks, flash memory cards, and other forms of storage media to uncover evidence of crimes” (Garber, 2001 as cited in Hall, 2005). This definition does not appreciate the preservation, collection, validation, identification, analysis, interpretation and presentation of digital evidence, which are important steps in digital forensics

ii. “Computer Forensics undertakes the post-mortem, or ‘after-the-event’ analysis of computer crime. Of particular importance is the requirement to successfully narrow the potentially large search space often presented to investigators of such crimes. This usually involves some form(s) of guided processing of the data collected as evidence in order to produce a shortlist of suspicious activities. Investigators can subsequently use this shortlist to examine related evidence in more detail” (Abraham & de Vel, 2002 as cited in Hall, 2005). This definition appreciates the manual role in digital forensics.

iii. “Computer forensics is the process of conducting an examination into the contents of the data on a computer system using state of the art techniques to determine if evidence exists that can aid in internal or legal investigations. Forensic specialists use a wide array of methods to discover data and recover deleted, encrypted, and damaged files” (Digital Forensics of Texas, 2005 as cited in Hall, 2005). This definition appreciates the role that digital forensics plays outside legal parlance.

iv. Computer forensics is “the investigation of computer crime, including the collection, analysis and presentation in court of electronic evidence. Also called: Cyber forensics, Forensics, Security Forensics, Digital Forensics, and Forensic Analysis” (Bitpipe, 2005 as cited in Hall, 2005).

v. Computer forensics can be defined as the set of “people, processes, tools and measures to gather, analyze and interpret digital data to support or refute certain allegations of misuse involving digital systems” (Cybex, 2005 as cited in Hall, 2005). This definition seems to have made from the management perspective of digital forensics.

vi. “Computer forensics is commonly described as the study of evidence derived from computers. However, the literature on the topic concentrates primarily on the recovery and preservation for presentation as evidence, of data from computers that may have been used in the commission of some criminal activity. In fact, most books on the subject focus on the preservation aspects, fastening on documentation of the chain of evidence, and almost ignoring the technical aspects of the field” (Slade, 2004 as cited in Hall, 2005).

vii. “Computer forensics is the application of scientifically proven methods to gather, process, interpret, and to use digital evidence to provide a conclusive description of cyber crime activities. Cyber forensics also includes the act of making digital data suitable for inclusion into a criminal investigation. Today cyber forensics is a term used in conjunction with law enforcement, and is offered as courses at many colleges and universities worldwide” (ISP Webopedia, 2005 as cited in Hall, 2005).
As indicated above, digital equipment may have an important role in traditional crimes also and thus digital forensics can be a necessity to establish traditional crimes too. For example: (1) the police may analyze email and Internet activities of murder and rape suspects to find evidence about their motives or hiding locations, (2) corporations usually investigate computers when an employee is suspected of unauthorized actions and (3) fraud investigations may require acquisition of transaction history evidence from servers (Carrier et al, 2003).

Digital forensics is, of course, highly technical, and therefore grounded in science: computer science, mathematics, physics, and so forth (Ryan et al, 2005). It is also a discipline that often requires knowledge of engineering, particularly electrical, mechanical and systems engineering. Applying the science and engineering in specific investigations is a complex process that requires professional judgment that is sometimes more art and common sense than science. The technical requirements of computer forensics are certainly rigorous and the forensic specialists should know a great deal about the normal functions of the operating system in question (Slade, 2004). Finally, computer specialists can draw on an array of methods, procedures and tests for discovering data that resides in a computer system, or recovering deleted, encrypted, or damaged file information. Using these methods, procedures and tests, digital forensic investigation is generally done on digital data to produce (legally accepted) digital evidence.

2.9.1 Steps in digital forensics

Digital forensics involves collection and documentation of valid digital evidence, and the validity often depends on its credibility. As Baryamureeba (2004) opines, every action to raise the credibility of digital evidence may strengthen its admissibility into the legal system. Credibility of digital evidence can largely depend on its integrity, which can be achieved mainly through proper acquisition and preservation of the evidence with the help of Standard Operating Procedures (SOPs) and standard forensic models. For instance, Reith et al (2002, p.6) suggests a digital forensic model. The key components of this model include the following:
a. Identification – recognizing an incident from indicators and determining its type. This is not explicitly within the field of forensics, but significant because it impacts other steps.

b. Preparation – preparing tools, techniques, search warrants, and monitoring authorizations and management support.

c. Approach strategy – dynamically formulating an approach based on potential impact on bystanders and the specific technology in question. The goal of the strategy should be to maximize the collection of untainted evidence while minimizing impact to the victim.

d. Preservation – isolate, secure and preserve the state of physical and digital evidence. This includes preventing people from using the digital device or allowing other electromagnetic devices to be used within an affected radius.

e. Collection – record the physical scene and duplicate digital evidence using standardized and accepted procedures.

f. Examination – in-depth systematic search of evidence relating to the suspected crime. This focuses on identifying and locating potential evidence, possibly within unconventional locations. Construct detailed documentation for analysis.

g. Analysis – determine significance, reconstruct fragments of data and draw conclusions based on evidence found. It may take several iterations of examination and analysis to support a crime theory. The distinction of analysis is that it may not require high technical skills to perform and thus more people can work on this case.

h. Presentation – summarize and provide explanation of conclusions. This should be written in a layperson’s terms using abstracted terminology. All abstracted terminology should reference the specific details.

i. Returning evidence – ensuring physical and digital property is returned to proper owner as well as determining how and what criminal evidence must be removed. Again not an explicit forensics step, however any model that seizes evidence rarely addresses this aspect.
The above model is an enhancement of the two previous models suggested by US Federal Bureau of Investigation (FBI) and the Digital Forensic Research Workshop (DFRW). The above model offers only a general framework and “additional sub-procedures would be necessary to define the different classes of digital technology under this model” (Reith et al, 2002, p.8).

There are many factors that might affect the integrity of digital evidence. One reason to cast doubt on (or at least occasionally suspect) the integrity of digital evidence is the very existence of a variety of tools and methods that allow almost anyone to modify almost any attribute associated with digital data (DFRWS, 2001). Another reason can be the lack of skill of the investigating expert and/or police to identify, collect and preserve digital evidence as part of digital forensics. In order to put more credence to digital evidence, SOPs have been introduced by the legislative and legal authorities of many countries. (Abe et al, 2005). These SOPs can be helpful not only to establish cyber crimes, but also traditional crimes because, as said above, digital evidence has often been found very effective and important even to establish traditional crimes.

To sum up, digital forensics can be considered as the application of computer examination and analysis techniques in the interests of determining potential legal evidence and is usually performed for the purpose of helping the judicial authorities by providing digital evidence to establish culpability. It is no surprise then that it has become a popular topic in the legal community (Li et al, 2003). Moreover, legal domain and judiciary appreciate that the digital forensics is highly technical, and therefore grounded in science (Ryan et al, 2005), which is outside their area of expertise.
2.10 **Technical experts and their role in digital forensics**

Any legal or judiciary system with a sense of responsibility and accountability would like to ensure that they be properly informed by experts in matters that are outside their area of expertise (United States Court of Appeals, 2001, p.20). A technical expert before the court of law is generally the person who is designated by the judge or any other law enforcement authorities to look into the technical aspects of a case and prepares a report that is legally convincing and binding. A technical expert thus needs to assist the judicial system to establish culpability in a manner that is transparent to non-technical persons as well, and help the judicial system in convincingly validating the legal process of resolving the situation. An attorney too can seek the help of an expert to perform a limited investigation with limited objectives, for which sometimes the expert must understand everything that has transpired (Kremen, 1998). Finally, because the judicial system has difficulties in mandating and interpreting standardization for computer forensics, it becomes the responsibility of the scientific community (i.e. the expert) to assist in this endeavor (Friedman et al, 1997). It is therefore the responsibility of the cyber forensic experts to assist the investigators, lawyers and jurors to detect, document and analyze digital evidence to establish culpability.

From the cyber forensic experts, the legal domain and judiciary demand (or solicit) only digital forensic evidence that is (1) “relevant”, (2) "derived by the scientific method" and (3) "supported by appropriate validation" (Ryan et al, 2005). The court in Daubert\(^\text{26}\) has suggested several factors to be considered to determine whether digital evidence possesses the requisite scientific validity. These are: (a) whether the theories and techniques employed by the scientific expert have been tested; (b) whether they have been subjected to peer review and publication; (c) whether the techniques employed by the expert have a known error rate; (d) whether they are subject to standards governing their application; and (e) whether the theories and techniques employed by the expert enjoy widespread acceptance. However, these factors are not exhaustive and do not constitute "a definitive checklist or test."\(^\text{27}\) The International High Tech Crime


\(^{27}\) ibid
Conference in 1999 adopted the following guidelines to achieve and preserve judicial admissibility of digital evidence: (i) Upon seizing digital evidence, no action should change that evidence; (ii) When it is necessary for a person to access original digital evidence, that person must be forensically competent; (iii) All activity relating to the seizure, access, storage or transfer of digital evidence must be fully documented, preserved and available for review; (iv) An individual is responsible for all actions taken with respect to digital evidence while the digital evidence is in their possession; (v) Any agency that is responsible for seizing, accessing, storing or transferring digital evidence is responsible for compliance with these principles (Strydom, 2001 as cited by Ryan et al, 2005, p.3). Computer forensic investigators must be aware of the legal environment in which they work, or they risk having the evidence they obtain being ruled inadmissible (Wegman, 2005, p.10).

Forensic examiners who do not account for error, uncertainty, and loss during their analysis may reach incorrect conclusions in the investigative stage and may find it harder to justify their assertions when cross-examined. In order to reduce their risk during cross-examination, the cyber forensic investigators should estimate how closely their measured values (results of investigation) represent the reality (Casey, 2002, p2). In network crime investigation, Casey (2002, p.6) observes that errors and uncertainties are either fabricated or unfortunate. He classifies the errors and uncertainties as (1) temporal uncertainty, for example, inaccuracy in the system clock of the computers involved, and (2) uncertainty in origin, for example, IP spoofing. Casey (2002, p.17) further observes that errors and losses can be introduced into log files at various stages: (a) At the time of the event (For example, a fake IP address can be inserted into a packet or a log entry can be fabricated to misdirect investigators); (b) During observation: (For example, spanning ports on a switch to monitor network traffic increases the load on the switch and may introduce loading error, increasing the number of dropped datagrams); (c) At the time of the log creation (For example, remote logging facilities such as syslog can have some percentage of lost messages and a clock offset on the logging system can result in incorrect date/time stamps); (d) After creation (For example, log files can become corrupted or can be altered to mislead investigators); (e) During examination (For example, the application used to represent log files can introduce errors by incorrectly
interpreting data or by failing to display certain details); and (f) During analysis (For example, mistakes by an examiner can result in incorrect conclusions). Several error detection methods, including probability distribution functions are often used to ascertain the error factor in the results of investigation.

There is also the issue of the qualification of the expert, formal or otherwise. On the qualifications of the expert, Meyers and Rogers (2004, p.10) observe that “The computer forensic field is fairly unique, as it has no credentials or a formal educational process. Currently, the lower courts accept qualifications based on the skills and previous work experience of the experts. While this has been sufficient to date, it is anticipated that contesting the expertise and qualifications of expert witnesses will become more common in the future. The need for a national and internationally recognized certification and standardization for computer forensics is necessary”. The expert needs to be a human being, at least in the US. The US courts have found that an inanimate object (e.g. a software package) cannot be considered an expert (Meyers and Rogers, 2004, p.5). (This may not mean that the object or results from that object cannot be used for scientific testimony. However, in some circumstances, the individual using the software package will have to attest to the procedures used).

2.11 Presentation of digital forensic investigation results

In digital forensics, researchers have described the importance of a standard, open format for digital evidence provenance, both for description and comparison of particular pieces of evidence, as well as for tool interoperability and validation (Levine et al., 2009). Moreover, Pladna (2008) proposed a standard digital evidence bag for a large organization to perform more efficient collection of data. Marcella et al. (2007) explained that a digital forensics laboratory accreditation standard, and standard operating procedure checklists are intended to act as guides to the uniform process of conducting digital forensics examination in a precise and accurate manner. The Common Evidence Format Working Group (2006) proposed defining a standard format for storing and transmitting digital evidence by using metadata so that it can be processed efficiently by multiple tools and parties, can ensure evidence integrity, and effective case management.
Garfinkel et al. (2006) designed a file format for a forensic image called the Advanced Forensics Format (AFF). This format is both open and extensible. Like the EnCase format, AFF stores the imaged disk as a series of pages or segments, allowing the image to be compressed for significant savings. Unlike EnCase, AFF allows metadata to be stored either inside the image file or in a separate, companion file.

Garfinkel et al. (2006) declared that, although AFF was specifically designed for use in projects involving hundreds or thousands of disk images, it works equally well for practitioners who work with just one or two images. Additionally, if the disk image is corrupted, AFF’s internal consistency checks are designed to allow the recovery of as much image data as possible (Garfinkel, et al., 2006).

2.12 Digital forensics against physical forensics

One important point remaining is the correlation of digital forensics with physical forensics. The basic building blocks of physical forensics can be of help in explaining digital forensics better. Brian Carrier and Eugene H. Spafford (Carrier et al, 2003) have examined the physical forensic models that could be applied to digital forensics as well. According to them, while physical forensics typically answers identification or comparison questions, such as "can substance X be identified" or "are substance X and Y the same", digital forensics is much more an involved process where the investigator must trace user activity and cannot provide a simple yes or no answer. They further note that the investigation of billions of bytes of digital data is similar to the investigation of a house where an investigator must look at thousands of objects, fibers, and surface areas and use his experience to identify potential evidence that should be sent to a lab for analysis. Their (Carrier and Spafford’s) model for digital forensics considers the computer as a separate crime scene and more than simply an object of physical evidence. Just as guns are sent for finger print analysis and the body is tested for chemicals in the blood stream, computers are analyzed to identify additional pieces of evidence and the sequence of events that occurred inside it.
2.13 **Software forensics**

Forensics of computer software is often referred to as software forensics and needless to say, is a part of digital forensics. Software Forensics is the area of software science aimed at authorship analysis of computer source code (Sallis et al, 1996). Identifying the author is usually only a part of the larger software piracy litigations and software forensics generally includes the forensics of malicious software (for example, virus) to establish the identity of the programmer and also the forensics of duplicated software (for example, software piracy) to establish intellectual property duplication.

Software forensics involves the analysis of evidence from program code itself. Program code can be reviewed for evidence of activity, function, and intention, as well as evidence of the software’s authorship and through software forensics, an expert can obtain evidence of intention (of interference) as well as the cultural and individual identity of the culprit (Slade, 2004). The process of software forensics is similar to medical forensics where the investigator examines the remains to obtain evidence about the factors involved in the crime (Spafford & Weeber, 1992).

With the increasingly pervasive nature of software systems, police cases can arise in which it is important to identify the author of a source code (Spafford and Weeber, 1992). Such situations include cyber attacks in the form of viruses, Trojan horses and logic bombs, fraud and credit card cloning, code authorship disputes, and intellectual property infringement (Chang-Tsun Li, 2010, p.470). While SCAP (See 2.14 and 2.15.2, below) helps to identify the author of malicious software, establishment of software intellectual property litigations requires procedures such as, the AFC test (See 2.17 below).

Software forensics generally requires source code analysis. Source code is the textual form of a computer program that is written by a computer programmer in a computer programming language and these programming languages can in some respects be treated as a form of language from a linguistic perspective, or more precisely as a series of languages of particular types, but within some common family (Gray, Sallis, & MacDonell, 1998). In other words, just as written text can be analyzed for evidence of authorship, computer programs can also be examined from a forensic or linguistic viewpoint for information regarding the program’s authorship.
2.14 **Software piracy forensics for tracing back to the author**

There have been several approaches using different techniques and theoretical frameworks to assess software piracy. Van der Ejik (1994) shows how Discourse Analysis, tracking patterns of semantically related elements in text, can be used to trace similarities cross-linguistically. But his findings are more theoretical and do not refer to its practical application in software forensics. Friedman et al. (1997) do refer specifically to the detection of the updated derivative from the original source code using the same programming language. The SCAP (Source Code Author Profiles) approach (Frantzeskou et al, 2007) is based on the byte-level n-gram profiles representing the source code author’s style, and extending a technique originally applied to natural language text authorship attribution. Another one, viz. the Software similarity Measurement Tool, SMAT, based on a similarity metric between two sets of source code files and proposed by Yamamoto (Yamamoto, 2004) has been successfully tested in BSD UNIX operating system. Lancaster & Culwin (Lancaster and Culwin, 2004) survey eleven source code plagiarism detection engines, including MOSS, used by the academic community to compare student submissions suspected of plagiarism (a kind of piracy) and classify them in terms of the programming language support and metrics used in them. The university plagiarism system, though not as judicial as the court of law, goes through a similar process of adjudication where the accused is given a right to defend themselves, if necessary even with the help of a legal expert.

2.15 **Authorship analysis**

Authorship analysis can sometimes be a part of software piracy investigation. Authorship analysis in natural language texts, including literary works has been widely debated for many years, and a large body of knowledge has been developed (Dauber, 1990). However, authorship analysis research in computer software is fairly recent, slightly different and comparatively difficult (Krsul, 1994). Although a text in programming language (popularly called source code) is much more restrictive in terms of syntax and
grammar than a text in natural language, there is still a large degree of (author-specific) flexibility when writing a program (Krsul and Spafford, 1995) and the general methodology of authorship attribution can be applied to texts in both natural and programming languages. Some of the researchers (Krsul and Spafford, 1995; MacDonell et al., 2001; Ding and Samadzadeh, 2004) in this field have adopted an authorship identification methodology that comprises two main steps. In the first step, the apparently relevant software metrics are extracted and in the second step, these metrics are used to develop models that are capable of discriminating between several authors, with the help of a statistical or machine learning algorithm (Chang-Tsun Li, 2010, p.471). To be more precise, when a set of writings of a number of authors is known, any given new piece of writing can be attributed to one of them and then a statistical hypothesis test can be performed to confirm or disprove this attribution. The essence of this method is identifying a set of features that remain relatively constant for a large number of writings created by the same person. Once a feature has been chosen, a given writing can be represented by an n-dimensional vector, where n is the total number of features. Given a set of pre-categorized vectors, one can apply many analytical techniques to determine the category of a new vector created based on a new piece of writing. Hence, the feature set and the analytical techniques may significantly affect the performance of authorship identification.

a. **Authorship analysis in natural languages**

The earlier studies (between 1887 and 1996) on natural language authorship attribution can be attributed generally to Mendenhall, Yule, Zipf, Mosteller, Wallace, Baayen, Burrows, Holes, Forsynthia, Tweedie, Elliot, Lowe, Matthews, and Merriam (Chang-Tsun Li, 2010, p.473). After 1996, the research in authorship attribution did pick up and acquire momentum when the syntactic analysis method of authorship identification (Chaski 1998, 2001) was scrutinized by a US federal judge in a Daubert hearing and its evidence was allowed into trial with full admissibility. Taking cue from the earlier studies as well as from the above referred judicial scrutiny, Stamatatos et al. (2001)

---

introduced a fully automatic method to extract syntax-related features which offered better performance compared to that of pure lexical-feature-based approaches. In addition, the proposed use of structural layout traits and other features for e-mail authorship identification by de Vel et al. (2001) also offered high performance in authorship identification. Subsequently, Chaski (2005) presented a computational and stylometric method which, according to Chang-Tsun Li (2010, p.473) has obtained 95% accuracy and has been successfully used in investigating and adjudicating several crimes involving digital evidence. The 90% accuracy of the n-gram language model (Keselj et al., 2003) on Greek newspaper articles, English documents and Chinese novels was a landmark in authorship identification across different languages. In this model, Keselj et al coined the term n-gram and defined it as an n-contiguous sequence that can be defined at the byte, character or word level. For example, the word sequence “On the” would be composed of the following byte-level n-grams (the character “_” stands for space).

<table>
<thead>
<tr>
<th></th>
<th>bi-grams</th>
<th>tri-grams</th>
<th>4-grams</th>
<th>5-grams</th>
<th>6-grams</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>On, n, _t, th, he</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>On, n_t, _th, the</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>On_t, n_th, _the</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>On_th, n_the</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>On_the</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Finally, in the n-gram language model, Keselj et al (2003) defines the author profile as “a set of length L of most-frequent n-grams with their normalized frequencies”. Thus, the author profile looks like a set of ordered pairs \{(x_1; f_1); (x_2; f_2)......(x_L; f_L)\}, which is mathematical by nature and thus, provides a way to easily analyze the author profile, mathematically. All the above methods were introduced for authorship analysis of literary works and not specifically for identifying the author of a computer software.

b. **Authorship analysis in software**

The history of authorship analysis in software can be said to date back to 1993 when Spafford and Weeber suggested (Spafford & Weeber, 1993) that it might be feasible to analyze the remnants of software (such as a virus, worm, or Trojan horse), and identify its author(s). By coining a new term *software forensics*, Spafford and Weeber (1992)
investigated and analysed both source code and executable code remnants and proposed that not only in the ordinary source and executable codes but even in the optimized executable code, there exist features that may be considered in the analysis: features such as data structures and algorithms, compiler and system information, programming skill and system knowledge, choice of system calls, errors etc. For source codes, they suggested analyzing features like programming language, use of language features, comment style, variable names, spelling, grammar etc. As an expansion of this work, Sallis et al. (1996) suggested some additional features such as cyclomatic complexity of the control flow and the use of layout conventions. Further, an automated approach was suggested by Krsul and Spafford (1995) to identify the author of a program written in C. Using around 50 software metrics (of three categories namely programming layout metrics, programming style metrics and programming structure metrics) and a software analyzer program, they extracted the features from 88 programs belonging to 29 programmers, eliminated the unimportant features found (and thus, arrived at a subset of metrics / features) and finally, by applying discriminant analysis (a statistical approach) on the chosen subset of metrics to classify the programs of authors, they achieved 73% overall accuracy. Subsequent works in this line of research include (1) a fuzzy logic approach on authorship of computer programs in C++ by Kilgour et al. (Kilgour et al, 1998, MacDonell et al., 2001); (2) a dictionary based system called IDENTIFIED (Integrated dictionary-based extraction of non-language-dependent token information for forensic identification, examination and discrimination) by Gray et al. (1998); (3) software metric based authorship identification of Java code by Ding & Samadzadeh (2004); (4) the proposed technique by Lange and Mancoridis (2007) in which code metrics are represented as histogram distributions; and finally, (5) the proposal of Kothari et. al (2007) to use a combination of style and text based metrics in order to represent each programmer’s style The latest in this list is the SCAP approach (Frantzeskou et al, 2007) which has its roots in the above referred n-gram language model (Keselj et al., 2003).

<table>
<thead>
<tr>
<th>3-gram</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>sio</td>
<td>28</td>
</tr>
<tr>
<td>_th</td>
<td>28</td>
</tr>
</tbody>
</table>
Table 2.1: N-gram frequencies extracted from a source code file (Frantzeskou et al, 2007)

| f_() | 20 |
| _=_  | 17 |
| usi  | 16 |
| _ms  | 16 |
| out  | 15 |
| ine  | 15 |
| 
/ | 15 |
| on_  | 14 |
| _in  | 14 |
| fp_  | 14 |
| the  | 14 |
| sg_  | 14 |
| _i_  | 14 |
| in_  | 14 |

SCAP approach, an n-gram approach in programming language authorship analysis, helps the investigator to identify the author of the source code in dispute and is an extension of n-gram model for natural language authorship analysis of Keselj et al.(2003) (Frantzeskou et al., 2007, Frantzeskou et al, 2008). A product of the team led by Georgia Frantzeskou of University of Aegeon, Greece, SCAP has been experimented on data sets of programs in different programming languages (Java, C++ and Common Lisp), made and modified by 6 to 30 candidate authors (For experiments and results, see Frantzeskou et al, 2005a, 2005b, 2006a, and 2006b). Results of these experiments helped Frantzeskou et al to claim that the SCAP approach is language independent and that SCAP’s n-gram author profiles can better represent the idiosyncrasies of the source code authors.

2.16 Features and parameters in software piracy research

Needless to say, software piracy research can be carried out along several parameters. In a very recent, thorough and comprehensive study, Holsapple et al. (2008) surveyed the parameters for software piracy research by “conducting a systematic examination of 75 articles dealing with software piracy”. The study largely provides “a theoretically grounded framework that identifies essential parameters to consider in studying and
understanding the phenomenon of software piracy” (p.214). The study, though revealing, essentially deals with an academic analysis of various psycho-social motivations of software piracy and their organization, and the results convey a “unified picture” of these motivations. Even while it contains a section on “Technological Guardianship” on the occurrence of software piracy that may be tangentially relevant to the work envisaged here (this thesis), none of the parameters in the article appears to address the procedural and the mechanical aspects of identifying and establishing software piracy or copyright infringement claims.

Unlike software piracy research, software piracy forensics research is often carried out along a different sets of parameters and features. Spafford and Weeber (1992) have listed out several features of the software along which forensic analysis of software should be carried out. According to them, software piracy forensics should be carried out on the executable code along parameters namely (1) data structures and algorithms; (2) compiler and system information; (3) programming skill and system knowledge; (4) choice of system calls; and (5) programming errors. Further, according to them, software piracy forensics should be carried out on source code along parameters namely; (a) language; (b) formatting; (c) special features like pragmas (or special macros), conditional compilation constructs etc; (d) comment styles; (e) variable names; (f) spelling and grammar; (g) use of language features (h) scoping; (i) execution paths; (j) programming bugs and (k) metrics. The list of measurements suggested by Spafford and Weeber is comprehensive, but the derivation of some of these is difficult to automate (Chang-Tsun Li, 2010, p.476).
<table>
<thead>
<tr>
<th>Measurement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>WHITE</td>
</tr>
<tr>
<td>2</td>
<td>SPACE-1</td>
</tr>
<tr>
<td>3</td>
<td>SPACE-2</td>
</tr>
<tr>
<td>4</td>
<td>SPACE-3</td>
</tr>
<tr>
<td>5</td>
<td>SPACE-4</td>
</tr>
<tr>
<td>6</td>
<td>LOCCHARS</td>
</tr>
<tr>
<td>7</td>
<td>CAPS</td>
</tr>
<tr>
<td>8</td>
<td>LOC</td>
</tr>
<tr>
<td>9</td>
<td>DBUGSYM</td>
</tr>
<tr>
<td>10</td>
<td>DBUGPRN</td>
</tr>
<tr>
<td>11</td>
<td>COM</td>
</tr>
<tr>
<td>12</td>
<td>INLCOM</td>
</tr>
<tr>
<td>13</td>
<td>ENDCOM</td>
</tr>
<tr>
<td>14</td>
<td>GOTO</td>
</tr>
<tr>
<td>15</td>
<td>COND-1</td>
</tr>
<tr>
<td>16</td>
<td>COND-2</td>
</tr>
<tr>
<td>17</td>
<td>COND-3</td>
</tr>
<tr>
<td>18</td>
<td>COND-4</td>
</tr>
<tr>
<td>19</td>
<td>COND-5</td>
</tr>
<tr>
<td>20</td>
<td>COND-6</td>
</tr>
<tr>
<td>21</td>
<td>COND</td>
</tr>
<tr>
<td>22</td>
<td>CCN</td>
</tr>
<tr>
<td>23</td>
<td>DEC-IF</td>
</tr>
<tr>
<td>24</td>
<td>DEC-SWITCH</td>
</tr>
<tr>
<td>25</td>
<td>DEC-WHILE</td>
</tr>
<tr>
<td>26</td>
<td>DEC</td>
</tr>
</tbody>
</table>

Table 2.2: Twenty six programming features of the IDENTIFIED (MacDonnel and Gray, 2001)
Authorship identification researchers suggest a different set of features along which the authorship identification research needs to be carried out. For example, Cannon et al. (1997) define coding style guidelines such as commenting, white space, declarations, naming conventions and file organisation. For experimenting the authorship identification techniques in c-language programs, Burrows et al (2009) create six classes of features of c-language namely white space, operators, literals, keywords, I/O words (from the stdio.h ANSI C89 header file) and function words (all standard library words not in stdio.h). A different analysis tool by name IDENTIFIED (MacDonnel and Gray, 2001) uses 26 different programming features (see Table-2.1 for the full list) to identify the author of a software.

Finally, Krsul and Spafford (1995) and Kilgour et al. (1998) used another set of quantitative metrics in order to classify C programs. They include; (1) Programming layout metrics (that measure indentation, placement of comments, placement of braces etc.); (2) Programming style metrics (that helps to measure character preferences and construct preferences and to find out statistical distribution of variable lengths and function name lengths etc.); and (3) Programming structure metrics (that include statistical distribution of lines of code per function, ratio of keywords per lines of code etc.). Measurements in these categories are automatically extracted from the source code using pattern matching algorithms. These metrics are primarily used in managing the software development process but many are transferable to authorship analysis (Chang-Tsun Li, 2010, p.476).

2.17 Software piracy and software copyright infringement

Authorship identification is only one of the objectives of software piracy forensics (say, in court cases that demand identifying the author of a specific virus). Software litigation may not always require authorship identification but instead, would come up in the court as an intellectual property violation litigation, in which case the objective of software forensics would be to prove or disprove copyright infringement.

One of the earliest approaches applied to infringement investigation of computer software was the “Iterative Approach,” which was highlighted in 1985 by a US district court in
E.F. Johnson Co., v. Uniden Corp. of America. To establish a prima facie case of infringement using this approach, it must be shown that (1) the defendant “used” the copyrighted work in preparing the alleged copy, which may be established by proof of access and similarity sufficient to reasonably infer use of the copyrighted work; and (2) that the defendant’s work is an iterative reproduction, that is, one produced by iterative or exact duplication of substantial portions of the copyrighted work. This approach seemingly has not seen widespread use, as the types of computer programs have became more sophisticated since the case was decided and other tests have been developed (Raysman and Brown, 2006).

To determine whether piracy (or copying) is actionable, the critical issue (in a court case alleging copyright infringement) is whether there is so-called “substantial similarity” between the defendant’s work and the protectable elements of the plaintiff’s work, a question that is not always easily answered in the computer software realm (Raysman et al, 2006). Often, a court case of such type makes or emphasizes the importance of methods like Abstraction-Filtration-Comparison (AFC) test for evaluating copyright infringement claims involving computer software.

AFC test was primarily developed by Randall Davis of the Massachusetts Institute of Technology for evaluating copyright infringement claims involving computer software and used in the 1992 Computer Associates vs. Altai case, in the court of appeal of the 2nd federal circuit in the United States (Walker, 1996). Since 1992, AFC has been recognized as a legal precedent for evaluating copyright infringement claims involving computer software in several appeal courts in the United States, including the fourth, tenth, eleventh and federal circuit courts of appeals (ESALab, 2007; Raysman et al, 2006; USDCM, 2010).

AFC test offers investigators, ways to investigate into various areas in the software namely functional area, reports generated, screens generated, menu structure, data structure, object code strings, documents, just to name a few (Davis, 1992, pp. 317-25, Kremen, 1998). AFC requires the investigator (1) to break down the plaintiff’s program into its constituent structural parts (“abstraction”); (2) examine each part for incorporated

---

29 623 FSupp 1485
“ideas,” elements taken from the public domain, methods of operation, processes or procedures, or otherwise unprotected material (“filtration”); and (3) compare the remaining kernel of creative expression, if any, with the pirated, at each level of abstraction (“comparison”).

Even though AFC is basically a manual comparison approach, there exists a software tool by name ‘SIMILE Workshop’, which was developed based on AFC’s theoretical framework, in 2007 by the French firm, European Software Analysis Laboratory (EsaLab) with additional inputs from Institut d’´electronique et d’informatique Gaspard-Monge – IGM, University of Marne-la-Vall’ee, France (ESALab, 2007). EsaLab claims that ‘SIMILE Workshop’ is the only software tool (available in the market) based on AFC. All these confirm AFC’s fitness both as a manual approach and as a theoretical foundation for a software tool for establishing copyright infringement claims involving computer software and so, no researcher (who works with the objective of formulating an advanced methodology for manually comparing two software packages) can avoid AFC.

The infallibility of the AFC approach has not stayed unquestioned. One of the best observations on AFC is the amicus brief (United States Court of Appeals, 1997) submitted to the United States court of appeals for the second circuit. This amicus brief was filed in 1996 by Mark M. Arkin, Counsel on behalf of five computer scientists, Roy Campbell (University of Illinois), Lee A. Hollaar (University of Utah), Randall Davis (the inventor of AFC), Gerald J. Sussman, and Hal Abelson, (all three of Massachusetts Institute of Technology), who believed that there was uncertainty among courts in how to implement the AFC for evaluating copyright infringement claims involving computer software. This legal document describes the technical complexity and difficulty of performing the AFC test “from the standpoint of computer science”, which these five computer scientists have experienced in their role as experts. The amicus brief observes that “Performing AFC test in the context of litigation is a challenging task, in no small

30 Abstraction-Filtration-Comparison Analysis Guidelines for Expert Witnesses, available at


31 See their brochure published in 2007 (ESALab, 2007)
measure because of the technical difficulties that arise. These technical difficulties stem from (a) the sheer magnitude of the task of analyzing programs that routinely consist of hundreds of thousands of lines of computer code; (b) the lack of any fixed or agreed-upon set of levels of abstraction by which to describe a program; (c) the interaction of legal doctrines (such as merger, scenes a faire, and public domain) with the technical constraints of the computer industry; and (d) the rapid evolution of these doctrines in the areas of computer software” (United States Court of Appeals 1997, p. 4).

Another observation (on AFC) worth mentioning is the one by honorable judge John M. Walker, Jr., (of United States Court of Appeals) through his article "Protectable 'Nuggets': Drawing the line between idea and expression in computer program copyright protection" (Walker, 1996). Through this article, the judge explains the difficulties in drawing the line between the idea and expression in software. By observing that the group of three doctrines (which is the core of AFC) “is not exclusive” and that “one might expect in future cases to see filtration occurring with respect to other program elements, including those listed in § 102(b) of the copyright act, such as procedures, process and methods of operation”, the judge fears that “even under an expanded list of elements excluded by the filtration step, there could still be many “nuggets” of protectable expression left in a program” (Walker, 1996, p.83) and because of this, the filtration stage is a battleground where the attorney who is able to argue successfully that his client’s program, after filtration, contains many elements of protected expression should be all the more likely to succeed on a claim of infringement. These observations of the judge also point to the limitation of the filtration stage.

As a modification to AFC, Dennis M. Carleton (1995) proposed a behavior-based test (called Identification-Filtration-Comparison test or IFC test) for determining software copyright infringement. This test starts with an analysis of the problem in the light of traditional copyright doctrine, the Copyright Act of 1976 and congressional intent. It then proceeds by analyzing how the courts have struggled with the concept of non-literal infringement of computer programs and finally modifies the Abstraction-Filtration-Comparison test based on his behavior-based test. The test proposed by him would
eliminate the "abstraction" step of the AFC test, which involves separating the levels of abstraction within the program's code, organization and structure. Instead, the behavior-based test would replace the abstraction step with an "identification" step, which dissects and identifies the elements of the program's behavior. The behavior-based test would eliminate much of the confusion and difficulty involved with applying the AFC test. Finally, Carleton applies the proposed test to the facts in Lotus Development Corp. v. Borland International to illustrate its advantages.

2.18 Technical Experts and their role in software copyright infringement investigation

Technical experts, as in any other cyber crime suits, play an important role in software copyright infringement analysis also. Nixon Peabody International LLP, a multinational firm of solicitors experienced in handling suits involving AFC, observes that “software copyright infringement analysis is complex, both in terms of the legal principles involved and the technical expertise necessary to conduct a proper review. Accordingly, technical experts play an indispensable role in assisting with such an analysis. For expert testimony to be useful (or even admissible), however, attorneys must educate the experts they retain in how to conduct an appropriate infringement analysis.”

Lee A. Hollaar (Hollaar, 2002) has emphasized the supplementary role of the expert (expertise, insight, commonsense etc.) in the interpretation of the AFC. Hollar, along with four other computer scientists (the 5-member court-appointed experts in Harbor Solutions v. Applied Systems, a software infringement suit) had suggested through an amicus brief that;

Qualified technical experts should play an important role in the abstraction process: Given the difficulty and complexity of creating a technically accurate set of abstractions, this court should clarify that this is properly a task for qualified experts in the field. As U. S. Court of Appeals Judge John Walker, author of the Altai decision, has noted, “Most juries, and most judges (myself included), are less than completely comfortable with the...


concepts and terminology of computer programs and need extensive education in order to make intelligent decisions.\footnote{Amicus Brief at 8-9 (citations omitted)}

Jon O. Newman (Newman, 1999), one of the judges on the panel (in the 2\textsuperscript{nd} circuit of U. S. judiciary) that received the above mentioned amicus brief indicates what he needs from expert witnesses in a trial or in the amicus brief on an appeal involving software copyright infringement:

These professionals would be well advised not to tell me simply that the source code is or is not protectable expression. Their opinion are relevant, but, as with all opinions, what renders them persuasive is not the vehemence of their assertion and not even the credentials of those asserting them; it is the cogency and persuasive force of the reasons they give for their respective positions. These reasons had better relate to the specifics of the computer field. For example, as Altai\footnote{982 F.2d 693, 23 USPQ2d 1241 (2\textsuperscript{nd} Circuit, 1992)} indicates, even with its overly structured mode of analysis, it will be very important for me to know whether the essential function being performed by the copyrighted program is a function that can be accomplished with a variety of source codes, which will strengthen the case for protection, or, on the other hand, is a function, capable of execution with very few variations in source code, or, variations of such triviality as to be disregarded, in which event protection will be unlikely. For me, this mode of analysis is essentially what in other contexts we call the merger doctrine – the expression is said to have merged with the idea because the idea can be expressed in such limited ways that protection of the plaintiff’s expression unduly risks protecting the idea itself.\footnote{Newman J. O. (1999) New Lyrics For An Old Melody, The Idea/Expression Dichotomy In The Computer Age, 17, Cardozo Arts & Ent. Law Journal, p.699-700}

\section*{2.19 Conclusion}

To sum up, the field of cyber forensics has experienced many changes in the recent years as technology that was on the drawing board yesterday is now part of everyday criminal investigation. (Williams, 2005). The authorities are often encouraged to invest more in (a) the research, (b) the training and certification of experts; (c) the required procedural infrastructure; (d) measures to include quality control and quality assurance; (e) establishing a national policy, legislation and standards and (f) assessing error potential through statistical analysis. One can sense an increased feeling among jurors that forensic
evidence may be unreliable unless it is backed by scientific methods and based on observed, provable / disprovable empirical data.