Chapter 1

Introduction

The overwhelming importance and proliferation of computers in the last few decades have no doubt created a massive industry, all under the umbrella term of ‘software’. The ever increasing importance and complexity of computer software has invariably made it one of the most influentially desired, and coveted commodity. Perhaps unsurprisingly, one of the most prominent side effects of this importance has been efforts to steal and/or appropriate software unethically and illegitimately: in other words, the emergence of software piracy or copyright infringement.

1.1 Software piracy as a cyber crime

Software piracy and copyright infringement are extensive probably because the job of copying software programs is easy. Software copyright infringement affects the global economy in general and software industry in particular. Moreover, they have a tremendous impact on the development of digital intellectual property and technology. The pervasiveness of the illegal copying of software is a worldwide phenomenon and such copying is generally considered as a (cyber) crime against global economy.

Software falls under the purview of intellectual property and thus, piracy of copyrighted software can lead to copyright infringement litigations. The copyright law in most countries generally prohibits illegal duplication of intellectual property. 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 act of piracy not only disregards the intellectual effort and
right of possessions of the author but also deprives the authors of the software of a fair return for their work (Thomas et al, 1990). The arbitration of software piracy and the interpretation of copyright laws are often a grey area. When called upon to arbitrate in matters of piracy, the judicial system solicits the support of technical experts who are typically required to substantiate issues arising from patent and copyright infringements, trade secret misappropriation, and software piracy.

1.2 Collecting evidence of software piracy

One of the most significant challenges of investigating any cyber criminal activity is obtaining all the evidence (Casey, 2002a, p5). Software piracy leaves behind digital evidence and evidence of software piracy crime is often obtained through software piracy forensics which is the process of investigating the crime using scientific tests or techniques. As a prelude to this forensics, the investigation agency obtains the original\(^2\) and the pirated\(^3\) software. The pirated software is generally made obtained by the police or judiciary through a seizure procedure or voluntary disclosure. The seizure procedure usually ensures that the tool touching an evidence drive must not modify the evidence drive in any way and must preserve all of the evidence on the drive (Casey, 2002a, p117). As the task of collecting the evidence requires technical expertise, the investigation agencies and the judge would often appoint impartial technical experts to establish piracy / non-piracy and to prepare a report that is legally convincing and binding.

\(^2\) Throughout this thesis, original means the version of the software that the complainant submits to the law enforcement agency for software piracy forensics. This thesis presupposes that the law enforcement agency has satisfactorily verified the legal aspects of the documentary evidence of copyright produced by the complainant and is convinced that the complainant is the copyright holder of this version of the alleged software.

\(^3\) Throughout this thesis, pirated means the allegedly pirated software
1.3 **Methods to assess software piracy**

After obtaining the original and the pirated, a full-fledged forensic investigation of the given code has to be done to establish software piracy. There have been several approaches, using different techniques and theoretical frameworks’ to assess software piracy.

Software piracy can be established either by way of authorship identification or by proving copyright infringement or both. Authorship identification methods include IDENTIFIED (MacDonnel and Gray, 2001), SMAT (Yamamoto, 2004), and SCAP (Frantzeskou et al, 2007), while a popular copyright infringement establishment methods is Abstraction-Filtration-Comparison (AFC) test (Raysman et al, 2006; USDCM, 2010). Authorship identification methods require tracing back through the source code to the authorship of the code. On the other hand, copyright infringement investigation requires comparing of two sets of programs and related files when one party lodges a complaint of copyright infringement against the other.

Authorship identification methods aim at identifying the author of the pirated. The basic assumption in any authorship identification is that when the linguistic features of a set of software authors is ‘known’, any given new software can be attributed to one of them through similarity of profiles. Further, such an attribution is usually confirmed or disproved through a statistical hypothesis test. Authorship identification methods generally are mathematical in nature and do require the investigator to possess or newly create a programming profile (user profile) of the plaintiff as well as of the defendant.

Most Authorship identification methods remain as academic products. Although SCAP (a product from Greece) and IDENTIFIED (a product from New Zealand) have strong mathematical backing, they are yet to be used popularly in or by judiciary system even in
their respective countries of origin and thus are yet to prove their judiciary-friendliness⁴. While SCAP is yet to be used even in a pre-judiciary but live situation, IDENTIFIED has already been put to use in two pre-judiciary but live situations in New Zealand.

Unlike authorship identification methods, copyright infringement investigation methods check whether there is so-called “substantial similarity” between the defendant’s work and the protectable elements of the plaintiff’s work. These methods differ from the authorship identification methods in that they are not mathematical by nature and they do not require author profile to establish copyright infringement. Instead, they establish the infringement or violation of copyrighted ideas and expressions by comparing two sets of software.

The leading method of analysis in cases involving allegations of nonliteral copying of either literal⁵ or nonliteral elements is the Abstraction-Filtration-Comparison (AFC) test first enunciated by the Second Circuit in the case Computer Associates V Altai⁶. This approach requires a court (1) to break down the plaintiff’s program into its constituent structural parts (“abstraction”); (2) to examine each part for incorporated “ideas,” elements taken from the public domain, methods of operation, processes or procedures, or otherwise unprotected material (“filtration”); and (3) to compare the remaining kernel of creative expression, if any, to the work alleged to infringe at each level of abstraction (“comparison”).

---

⁴ E-mail communication with Dr. Stephen MacDonnel, the author of IDENTIFIED

⁵ In cases involving literal copying, expert analysis may be necessary to assist in establishing that the plaintiff’s software (or material portions thereof) was literally copied. A simple side-by-side comparison of the plaintiff’s work and the work that it is alleged to infringe will suffice. Experts are frequently useful in pointing out programming errors shared by both programs that logically could have arisen only as a result of literal copying. This analysis may be more complicated in cases involving literal copying of nonliteral elements (such as a menu-command hierarchy). Still, all that is required to establish copying in such cases is a side-by-side comparison of the nonliteral elements of the plaintiff’s work and the allegedly infringing work

The table below summarizes the typical allegations in software copyright infringement cases and the type of analysis required:

<table>
<thead>
<tr>
<th>Literal Copying of Literal Elements</th>
<th>Literal Copying of Nonliteral Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Side-by-side comparison of source code and “filtration” analysis)</td>
<td>(Side-by-side comparison of nonliteral elements and “filtration” analysis)</td>
</tr>
<tr>
<td>Nonliteral Copying of Literal Elements</td>
<td>Nonliteral Copying of Nonliteral Elements</td>
</tr>
<tr>
<td>(Abstraction-filtration-comparison analysis)</td>
<td>(Abstraction-filtration-comparison analysis)</td>
</tr>
</tbody>
</table>

Table 1.1: Types of analysis required in the investigation of a typical software copyright infringement

Among all the above mentioned authorship identification and software copyright infringement investigation procedures, Abstraction-Filtration-Comparison (AFC) test is the oldest (1992) and seems to be the most successful and widely used techno-legal tool for establishing software copyright infringement. AFC has an excellent track record of usage in US judiciary even since its inception in early 1990s. Although various AFC-like tests (or slight variations of AFC) are used in copyright infringement cases, depending upon the jurisdiction of the court, the AFC test procedure that was used in the Altai test\(^7\), in addition to binding district courts in the Second Circuit\(^8\), seems to have emerged as the preferred one with several other circuits having adopted its analysis or discussed it with approval.

\(^7\) Computer Associates International, Inc. v. Altai, Inc., 982 F.2d 693, 702 (2d Cir. 1992), US
\(^8\) See e.g., eScholar, LLC v. Otis Educ. Sys., 2005 U.S. Dist. LEXIS 40727 (SDNY Nov. 3, 2005)
1.4 Quality of the piracy investigation report

The technical expert’s report needs to be as thorough, authentic and convincing as possible in order to avoid legal challenges to the report and the resulting delay in justice. Although many automated forensic tools are available to help the technical experts to carry out their software piracy forensic assignment, the inherent complexity in software programming logic and underlying global commonalities in software demand a vital role for cyber forensic expert’s intelligence, expertise, common sense, insight and perhaps even intuition.

Further, the expert’s evidence is always open to legal challenge and such challenges, irrespective of the outcome, might delay the process of litigation and deny justice to the innocent. So, it is important for the expert to preempt such delay and denial by making the report as authentic, comprehensive and complete as possible. This might require supplementing the software intelligence with human intelligence, insight and common sense so as to produce and present a report as unchallengeable as possible. Hence, it is both crucial and necessary for the expert’s report to be as thorough, authentic and convincing as possible in the interest of proper justice.

1.5 Contention of this thesis

Today’s software world features users and developers who are more intelligent, better-informed, more techno-savvy and better-equipped. Moreover, criminals are also expanding their horizons into the software space. This being so, it is the duty of computer experts to update the software forensic tools, processes and procedures so as to make tools, processes and procedures fit for performing the forensics of modern software products rather than letting legal professionals lie back complacently on the legacy of
forensic tools like AFC (more than two decades old). When forensic tools are old and outdated, the results of forensic analysis might be incomplete and inadequate.

This thesis contends that new developments in software technology has widened the scope and opened up further space for a comprehensive method for manually comparing two software packages by applying human intelligence, insight and common sense in addition to the expert’s expertise. Manual comparison, which supplements and not replaces automated comparison, would give the expert scope to add value to the result. Such a value-added report is what a judge normally expects from a technical expert. This thesis argues that such a manual method of comparing two software packages needs to address not only comparison-issues related to various parts of the software like source code, object code, databases, and fingerprints (like most existing tools do) but also other piracy related issues like post-piracy modifications, design patterns and data piracy. The thesis therefore takes upon itself the task of proposing, formulating and empirically testing a manual comparison procedure which is fundamentally a modified form of Abstraction-Filtration-Comparison (AFC).

1.6 Tasks performed in this research

As part of creating POSAR, the software piracy forensic protocol as an alternative to (as a modified form of) AFC, the following tasks were involved.

i. Defining the set of parameters along which the piracy forensics of a Management Information System (MIS) can be carried out. These are parameters concerning the stylistic, structural, compositional, logical, taxonomic and nomenclature features of MIS. Moreover, parameters concerning other features like special co-relation of irregularities (errors, blunders) have also been defined. These findings have already been published.

ii. Defining the set of parameters along which the potential post-piracy modifications are to be investigated. The logic of the argument is that the observed surface differences between the original and pirated does not necessarily provide automatic grounds for
exculpation from piracy in that many times much of the observed differences could be a direct result of post-piracy modifications both in the original and the pirated. The findings, mainly concerning the investigation of potential post-piracy modifications in MIS, have already been published.

iii. Identifying the scope and space for a comprehensive method for manually comparing two software packages. These findings have already been published.

iv. Defining a set of parameters for modifying the current procedure, Abstraction-Filtration-Comparison (AFC), which is already in use in the United States judiciary since 1992, for establishment of software copyright infringement. Also identified the areas of AFC that do not need modifications. All these new set of parameters have already been published.

v. Defining the term programming blunder and established its forensic importance. The contention here is that this particular phenomenon of blunder needs to be studied systematically from their very genetic origins to their surface realizations in contrast to bugs and flaws, especially in view of their importance in software copyright infringement forensics and this has already been done. Some suggestions as to their applicability and functional importance for cyber forensics are also given.

vi. Modifying the existing form of AFC. In order to make the modified procedure convenient for practitioners to use, an algorithmic form of it has been formed, which is the ultimate objective of this research. This modified procedure has been christened as POSAR (Planning-Operationalization-Separation-Analysis-Reporting) procedure. Just as AFC, POSAR also has a subjective procedure.

vii. Performing piracy forensics on a select set of ‘original’ and ‘pirated’ programs (including code segments, data bases etc.) using POSAR as well as AFC and further compared the forensic results to find out the effectiveness of POSAR over AFC.
1.7 **Pragmatics of this research**

This work envisages software piracy as a cyber crime and looks into what cyber forensic experts need to know while assisting the law enforcement agencies to establish culpability in a software piracy suit, particularly while using AFC. Generally, software piracy forensic experts are expected to work with an objective of carrying out software piracy forensics in a manner that is transparent to non-technical persons as well. The results of forensics investigation are often intended for use by the law enforcement officials (including jurors) who cannot be expected to be techno-savvy. This being so, this work looks for a modified way of AFC in a pragmatic way to make AFC’s process and result judiciary-friendly. Although this work involves theoretical analysis, what is envisaged is not just development of theory but of pragmatic procedural aspects for establishing guidelines and protocols to the legal system to establish and to detect software piracy (or to establish software copyright infringement) using AFC.

1.8 **Chapter formulation**

This thesis has been organized as follows.

Chapter 2 of this thesis tries a general overview of the field of cyber forensics and reviews the main studies in this field thereby situating this work properly. With the backing and support of the main works done in the various fields, the chapter discusses the various aspects of cyber crime, cyber forensics, digital evidence, software piracy, software copyright infringement, the related forensics including authorship analysis (starting with its root in linguistics), software copyright laws and litigation process and the role of the cyber crime expert specifically in the judiciary process.

Chapter 3 does mainly four things. First, it lays out and justifies the methodology adopted for this study and the data used. Next, it shows how AFC is in a state of readiness for modification and looks into what features of AFC are qualified to remain intact and what
features need to be modified. Thirdly, it shows how a qualitative study would be the best fit for modifying it. Finally, it also formulates some developments that form prime motivations for this specific research.

Certain developments that inherently and essentially need consideration while investigating software piracy do not appear to have been accessible through AFC. Any discussion on the modification of AFC necessarily would need to clearly articulate these. Some of these are elaborated in Chapter 4, 5 and 6. These are stated and explained here because a proper a consideration and understanding of these factors contribute to a better understanding of what ensues in chapter 7.

Chapter 4 of this thesis argues how a proper study of post-piracy modification of the pirated will contribute substantially to the reliability of software piracy forensic investigation. This chapter attempts to discuss the impact and implications of post-piracy modifications in software piracy forensics (or software copyright infringement investigation) and to recommend the need for proper amendments to be made in the existing forensic approaches / techniques (especially in AFC) so that evidence concerning post-piracy modifications gets proper consideration and treatment.

Chapter 5 is an attempt to explore the interesting point of how to deal with the increasing use of design and programming patterns in general software piracy forensics and in AFC in particular. The ultimate objective of this chapter is to discuss this point, in the context of the limitations of AFC. It also proposes some judiciary-friendly formats for presenting results to judicial officers.

One of the core design pattern / programming pattern-related issues is how to deal with programming blunder (Baboo and Bhattathiripad, 2009) and chapter 6 discusses this issue in detail with the primary objective of modifying AFC along this dimension. This issue has been discussed in this chapter right from definitional level of programming blunder
up to the level of its cyber forensic relevance since the area of programming blunders is undefined in software piracy and copyright infringement forensics.

The quest for modification for the AFC protocol in the form of an enhanced variation is the main objective of chapter 7. The new enhanced protocol, named as POSAR, is explained as consisting of five phases and they are: (1) Planning phase; (2) Operationalization phase; (3) Separation phase; (4) Analysis phase and; (5) Reporting phase. POSAR reformulates the software piracy forensic (copyright infringement investigation) procedure from the old 3-stage, linear sequential AFC to a 5-stage, cyclic, and structured process.

Establishing the effectiveness of POSAR test over AFC test empirically is the matter of concern in chapter 8. The effectiveness of POSAR over AFC has been established by applying both in test cases drawn from real life software piracy litigations and then comparing the results. Here, for sampling the test cases and for ensuring the reliability of results, the researcher’s own experience (as the cyber forensic expert in software infringement cases) has been utilized.

Chapter 9 discusses the implications of the results from Chapters 4, 5, 6, 7, and 8 and thus establishes the overall effectiveness of Planning-Operationalisation-Separation-Analysis-Reporting (POSAR) test over Abstraction-Filtration-Comparison (AFC) test in establishing software piracy / copyright infringement. It concludes with suggestions for future research and scholarly activity, particularly in the area of customization of POSAR.

1.9 Conclusion

To summarize, this work is primarily an attempt to show how a dedicated expert can supplement his/her intuitive expertise with objective empirical evidence based on various
elements that can be found in the original and the pirated (and that inherently and essentially need consideration in such a forensic task), to a judicially convincing level. Secondly, this work brings to light the forensic importance of certain tangible elements of software commonalities that may well be dismissed as insignificant or predictable by the existing software piracy / copyright infringement investigation methods. Thirdly, this work shows that how systematically, algorithmically and easily, the software piracy forensic expert can perform his / her forensic task and establish piracy / non-piracy using the newly formulated POSAR test. Fourthly, this work proposes a few formats (for presenting the result of software piracy forensic task) which are better convenient to technical experts as well as judicial officers.

Finally, this work arises from and is motivated by several years of practical experience in the courts of law in India by the researcher as technical expert in many software piracy / copyright infringement litigations.