CHAPTER I

INTRODUCTION

The legal scheme of copyright is presumptive of the need to extend protection to new and evolving forms of art and creativity. It is needless to mention that a complex technology like computer program has often juggled hard to adapt itself to the copyright law framework. After long and prolific deliberations on the legal and policy implications of extending copyright protection to computer programs, it is now well established that it has grown as a distinct sub-discipline in copyright system across the world. Copyright protection for computer programs has sound juridical basis in international \(^1\) and national copyright law. The Indian law on copyright –The Copyright Act 1957 (as amended up to date) specifically extends copyright protection to computer programs as a form of literary work.\(^2\)

A programmer while developing a program has to work under constraints of programming language and techniques that may fall within the continuum of ideas, expression and algorithms.\(^3\) While ideas are not protected, expressions are. The scope of what falls within the protectable expression and unprotectable idea is resolved by the application of idea-expression dichotomy. Copyright protects originality in expression. However, it also extends protection to non-literal elements in literary works. This presents various challenges in discerning ideas from original expressions. Protection for computer programs can embody the interface within it.

---

1 The International mandate governing copyright protection as provided in the Agreement on Trade Related Aspects of Intellectual Property Rights (TRIPS Agreement) has explicitly mandated under Art.10.1 that “Member countries shall extend copyright protection to computer programs as a literary work under the Berne Convention (1971), whether in source code or object code”. It must be noted that the Berne Convention itself does not explicit about it, but Article 2 of the Convention provides for a non-exhaustive list of kinds of works to which copyright protection might extend. Thus computer programs that satisfy all the requirement of copyright law viz., originality, fixation etc. is arguably protected under the Berne Convention (1971).

2 Sec 2(o) of the Copyright Act 1957 states that, “literary works includes computer programme, tables and compilations including computer [databases]”; sec 13 states that only “original literary works” are protected. But nowhere in the Act has originality being defined.

Such interfaces (both user and functional) are copyrightable written expressions and utilitarian functionality (ideas). Such functionality must be easily accessible to all since they form ideas in a program. The Indian courts have invoked the idea-expression dichotomy from time to time as having doctrinal basis in the Indian copyright system.\(^4\)

While the debate on the rationale of extending copyright protection to computer programs seems to have fairly settled\(^5\), there still exists a murky legal area concerning the nature, scope and extent of fair dealing of computer programs.\(^6\) Much of the concern is due to the nature of innovation in software technology and the compatibility requirements of computer program, technically referred to as achieving interoperability.\(^7\) This essentially harps on an important point that computer program are utilitarian works and hence face special problems. As functionality in a computer programme is not protected by copyright law. Computer program that de facto cover functionality must be allowed for analyzing their underlying idea. The fact that the object code (which cannot be understood by man) is protected through copyright, determining ideas/function or achieving


\(^{6}\) Limitations and exceptions to copyright are specifically allowed under the TRIPS Agreement. Art.13 TRIPs Agreement states: “Members shall confine limitations and exceptions to exclusive rights to certain special cases which do not conflict with a normal exploitation of the work and do not unreasonably prejudice the legitimate interest of the right holder”. Art 13 has been interpreted by a WTO Panel in a dispute between European Communities and the United States in the year 2000. However, it should be noted that the WTO Panel’s decision was limited to the application of the three-step test as part of Art 13 of the TRIPS Agreement.

\(^{7}\) There is no agreed definition as to what constitutes “interoperability” and how much of it is good for innovation and compatibility. There is a view that “interoperability” can be horizontal and vertical. “Horizontal interoperability means the availability of interface information that allows the reverse engineering to develop his own operating system while being compatible with already existing application software”. “Vertical interoperability is accomplished by software providers by disclosing interface information for their given operating system to application software providers to the extent to enable them to create new user programs independently”. See, Petra Heindl, “A Status Report from the Software Decompilation Battle: A Source of Sores for Software Copyright Owners in the United States and the European Union?” *TTLF Working Paper* No. 1, pp. 35-37. Available http://www. law. stanford. edu/ program/ centers/ ttlf/ papers/ heindl_ wpl.pdf (last visited on 20-12-2015).
interoperability through the process of reverse engineering without incurring the guilt of infringement is impossible. Much of the concern regarding decompiling emanates due to intermediate copying of a program. The reproduction, adaptation and translation rights accruing to the author are generally in question. Reverse engineering methods involve copying of the copyrighted computer program and the question is if such intermediate copying as a procedure is legally sustainable in consequential to the final outcome. Thus intermediate copying (unless excused for specific purposes) is violative of the reproduction right of the copyright holder. Intermediate copying during reverse engineering, however is not straightforward copying since the output of the reverse engineering code may be non-infringing and hence should be allowed under the theory of fair use. Further, the economics of innovation in the software industry demands that reverse engineering through decompilation should not be prohibited. Some have even argued that even if reverse engineering does not lead to additional innovation by building competing products it does provide consumer with an equal choice at a competitive price and hence enhances consumer welfare.

10 Both in the EU and the US, the fact the copyright code (object and source) is protected as expressions, making a copy or adaptation or both of the reconstructed source code, whether only a single one, or merely a preliminary one to further uses rather than a final one, will constitute infringement of the copyrighted computer program. See MAI System Corp. v. Peak Computer Inc., 991 F.2d 511, 519 (9th Cir. 1993). The EU position, as mentioned in the directive is that “loading, displaying, running, transmission or storage of the computer program” to require potentially infringing reproduction.
11 Terril Lewis, “Reverse Engineering of Software: An Assessment of the Legality of Intermediate Copying”, 20 LOY. LA. ENT. L. REV. 561, 563–4 (2000). As reverse engineering through contentious processes like decompilation prima facie violates the right of the copyright owner, it is important to see if at least such act can be legalized by looking at the purpose. Reverse engineering within the program context is carried out for various reasons, which may both be with a commercial or non-commercial intent including, building of a competitive product (which can be substitute), achieving interoperability, and research or error fixation.
13 Ibid. Among various reasons, developing a competing program appears to be most common in the industrial context.
Computer program technology presents a strong possibility of swathing unprotected elements of a program that are beyond the scope of copyright and hence must be available in the public domain. This necessitates access through principles of fair use. Consequently, the copyright act 1957, itself provides for certain exceptions in the nature of reverse engineering and other limited exception. However, private ordering through contract law that imposes restrictions on fair dealing/use can undermine the essential balance that the copyright law exceptions provide. Further, with the emergence of technology protection measures (TPMs) as an accessory to the copyright law, there are unique challenges in evaluating and understanding exceptions concerning reverse engineering of computer programs in the Digital Right Management Context. If recent judicial trends in comparative jurisdictions were to be placed emphasis upon, certain excesses that traditionally encompassed the rights granted under copyright and hence beyond the scope of ex ante realm of fair dealing, are now being addressed through ex post treatment competition law and policy. Thus it wouldn’t be a hyperbole to state that public access to copyrighted works is one of the peremptory principles of copyright law, and the challenges lies in balancing competing interests by clearly articulating the public interest goals of the copyright system.

India has included the scheme of reverse engineering for interoperability and program analysis within the scheme of fair use. However, the Indian Act does not define any such words within the Act- a trait of the common law tradition.

1. **Reverse Engineering of Computer Program Defined**

The concept of reverse engineering as applied to computer program normally refers to a variety of practices undertaken to understand how a computer

---

14 Section 52(1) (aa), (ab), (ac), (ad).
15 Sec 65 A of the Copyright Amendment Act 2012.
16 In many ways, evaluating such competition law principles that have emerged through case laws and internalizing them into the copyright fair dealing schema can be of good help in interpreting the exact notion of fair dealing of computer programme under copyright. See, for e.g., United States v. Microsoft, 65 F. Supp. 2d 1 (D.D.C. 1999); Judgment of the European Court of First Instances (Grand Chamber) of 17 September 2007.
program is built and how it achieves its functionality. Unlike other forms of literary expression, such as books, program cannot be simply “opened up” and read or examined. In its finished state (“object code” form), computer program consists of machine-readable object code that is not meaningful to or comprehensible by humans. Most software is sold or leased to end users in object code form. While software users can easily observe the outward functioning of the program, they cannot as easily perceive the ideas, processes, structures, or actual methods of operation of the program as it was written.

In the typical software development process, programmers write code in a programming language using alphanumeric characters that can be understood by a person familiar with the language. This form of the program is referred to as “source code.” After the source code is written, it is translated by a “compiler” program into the machine-readable object code.

In order to understand the ideas and “inner workings” of a computer program, one must therefore obtain either the original source code or detailed written specifications from the program’s developer. If these cannot be obtained, it becomes necessary to undertake a process of independently “decompiling” the object code back into source code. Because of factors inherent in the present technology, it is practically impossible to decompile object code back into an exact replica of the original source code.

For the purposes of copyright law analysis, it is important to understand that it is impossible to undertake the process of decompilation without at some point making a copy of either some or the entire program. Copying may take the forms of loading the program into computer memory, outputting it to a screen or printer, or copying it to other media.  

17Sega Enterprises Ltd. v. Accolade, Inc. 977 F.2d 1510 (9th Cir. 1992).
The information provided by decompilation of software can be used for a variety of purposes. Briefly, the objectives of decompilation might be categorized as follows\textsuperscript{18}:

- Copies. Information regarding the structure, functions, ideas and expression embodied in a program may be sought for the purpose of creating an identical or substantially similar substitute product.

- Functional Equivalents. Alternatively, the information could be sought for the purpose of creating a functionally equivalent, but not identical, product. The distinction between this type and the direct copy is that the developer of the new code aims to enable his program to perform the same function as the code that has been studied, but achieves that functionality by code structure and procedures developed independently (not copied) from the original code.

- Interoperable Products. In addition to copies or functional equivalents, information about a program is frequently desired in order to build interoperable software or hardware products, or to provide service. Such interoperable products could enhance or add entirely new functionality to the original software. The need for such information is particularly great when the software has obtained widespread acceptance in the market, has become a standard, represents a critical component in a larger system, or provides “low level” and highly functional services in a system.

2. **Significance**

Technology is evolving everyday as consumers spend countless amounts of money buying new products and companies compete to produce better products. One catalyst of this technological innovation is reverse engineering by both

developers and consumers. Reverse engineering is a method of recreating existing engineering concepts by analyzing the design and components of a final product to ascertain how the product operates. Although this is clearly distinguishable from the traditional concept of forward engineering- which requires creating a product from abstract engineering ideas and concepts- it has been practiced as a useful tool to learn how to build a technology and make improvements. Reverse engineering is well – exemplified in the computer software industry, where programmers constantly examine existing software to better understand the structure and make improvements on its operability.19

Reverse engineering is an effective tool to drive competition and innovation, to promote a greater good for consumers and the public at large. One way to do this is by providing more flexible interoperability exceptions for reverse engineering to expand choices and reduce cost for consumers.

3. **Review of the Literature**

In the light of significance of the study, the researcher has surveyed available literature on the subject as represented below:


Earlier literature was focused on the scope of copyright protection of computer program. Addressing the issue that copyright protection is a weak law in protecting utilitarian work like computer program. Computer program like any other

---


There is not much study in this area done in India. Except a few like, ArathiAshok, “Technology Protection Measures and the Indian Copyright (Amendment) Act 2012: A Comment”, 17 (6) J.Intellect.Prop.Rights 521 (2012).The article looks into the extent of protection that technological measures enjoy under the Indian Copyright Act 1957 and their shortcomings and to what

The author examines the Indian Law on Legal Protection of Software and takes stock of the types of software contracts and the nature of licenses that are generally entered into by the parties.

S. K. Verma, “IP Protection of Software and Software Contracts in India: A Legal Quagmire”, 17(4) *JIPR* 284 (2012). In this article, the author examines the Indian Law on Legal protection of software and takes stock of the types of software contracts and the nature of licenses that are generally entered into by the parties. But the author is silent on the legal validity of software license prohibiting reverse engineering of computer program.

Zakir Thomas, “Digital Technologies and Emerging Copyright Scenario”, 8 (7) *JIPR* 276 (2003). The article explores the consequences of DMCA, the impact on fair use and on the market place in general. Further examines the concerns of the developing countries in securing access to information and the suggestions of the Commission on Intellectual Property Rights. Noting the importance of Copyright as a public policy tool, the author pleads for calibration of the copyright balance to suit India’s national interest.

All these studies are focused either on copyright or anti circumvention law independently in developed nations. As far as Indian position is concerned, the research is limited to the scope of copyright protection of computer program; the scope of reverse engineering under the copyright law, there is neither case law nor any literature which deals with the contractual validity of software license term.
prohibiting reverse engineering. Further, with respect to Copyright Amendment Act 2012 introduced Sec 65 A to prevent circumvention of TPM applied by the copyright owner to prevent access to the copyright work. There are studies which discuss the impact of sec 65 A on fair use but not in specific to reverse engineering of computer program. None of the studies have focused on the interplay among various laws in relation to reverse engineering of computer program. Hence, this necessitated the researcher to make a critical analysis of these laws in comparison with developed countries.

4. **Statement of the problem**

Reverse engineering right is necessary to strike a proper balance between providing adequate copyright protection and promoting public access to a copyrighted work. A major fair-access issue has arisen in situations where a copyright holder attempts to restrict a reverse engineering right by copyright infringement litigation, contractual shrink-wrap licenses, or protective system even though the reverse engineering is permissible under the “fair use” doctrine. Further, if a software vendor that has market power in the primary market intends to eliminate existing or potential competition by unilaterally imposing anti-reverse engineering terms, and in doing so brings about anti-competitive effects in the relevant markets. Importantly software industry represents strong network effects. One convincing argument is that network effects support permitting a reverse engineering right in order to achieve compatibility with a dominant entity since dynamic competition benefits from network effects, which will increase consumer welfare.

The researcher has selected the topic “**Reverse Engineering of Copyrighted Computer Program: A Comparative Study**” for the avowed object of propose a clarification of the reverse engineering right, which policymakers should draw in terms of copyright, contract and competition laws.
5. **Objectives of the Study**

The specific objectives of the study are:

1. To study the scope of copyright protection of computer program.
2. To critically analyze the TRIPS Agreement and legal provisions in EU, US and India on computer program and competition.
3. To understand the relevance of reverse engineering in the development of compatible or interoperable program, competitive program and enhancement of programs.
4. To examine the provisions concerning fair use/dealing of computer program in EU, US and in India and critically evaluate its nature and scope.
5. To study the impact of Anti-Circumvention law and the scope of Competition Act on the reverse engineering of computer program.
6. To examine the legal validity of software license [EULA] prohibiting reverse engineering.
7. To critically appraise the judicial decision on idea-expression dichotomy in computer program.
8. To study the doctrine of ‘copyright misuse’ and to understand the ‘essential facility’ doctrine and its relevance to prevent the abuse of monopoly right by copyright owner.

6. **Hypotheses**

The following hypotheses are formulated for the purpose of the study:

1. Copyright protection of computer program conferring on the copyright owner a *de-facto* monopoly on functional aspects result in patent like protection.
2. Reverse engineering serving valuable social purposes may pave the way for protection from liability.

3. The impact of the Anti-Circumvention Law on non-infringing uses of copyrighted work leads to friction in the objective of copyright law.

4. Contract terms aiming to override the decompilation privilege would prevent the innovation in software industry.

5. The anti-competitive clause in software license agreement results in copyright misuse.

6. Copyright protections of computer programs were designed to promote competition to restraint the dominant company monopolizing in the relevant market.

7. **Methodology**

   Since the study is doctrinal base, the researcher relied on primary and secondary sources like international instruments, law, books, journal, and web source. Analytical method is used for the critical analysis of copyright, anti-circumvention, and competition law to reach the goal. Further, the researcher made an analysis of the international instruments of US, EU and India on computer program by employing comparative method.

8. **Scope and Limitations of the Study**

   The study is confined to copyright protections of computer program and that to restrict to computer software distributed only in its unintelligible object code, so called “closed software”. The researcher set aside open source software\(^\text{20}\) for the

\(^{20}\) “Open source” software lies in disclosure of the source code. The source code embodies copyrightable as well as non-protectable component of a computer program. Disclosing and thereby enabling anyone to copy source code thus may turn out to be a software copyright proprietor’s “open sores”. 
purpose of research. In the techniques of decompilation (or disassembly) or conduct an analytical “clean room” operation process, a program’s object code is first copied and then transformed into a human readable computer code, the source code. In addition to intermediate copying of the original computer program (decompilation), the process of reverse engineering also involves building the source code program based on the decompiled information of the original object code (implementation). Decompiling object code produces an approximation of the original source code, known as “intermediate copy”.

9. Scheme of the Study

The study is divided into seven chapters. The introductory chapter contains the need for study, review of literature, clear statement of the problem, objectives of the study, hypotheses, methodology adopted, scope and limitations of the study.

The second chapter “Reverse Engineering of Computer Program” deals with the computer program technology and the significance of compatibility. The subject matter scope of copyright protection of computer program is also examined in this chapter. Further, the researcher examines the relevance of reverse engineering in the software industry.

The third chapter “The New Phase of Fair Use: Accessing Copyrighted Programs through Reverse Engineering” discusses the doctrine of fair use and examines to what extent reverse engineering of computer program is permitted as a fair use under different jurisdiction such as US, EU and India.

The fourth chapter “Impact of Anti Circumvention Law on Reverse engineering” deals with the impact of anti circumvention law on reverse engineering of computer program, with a focus on DMCA, EU Copyright Directive and Sec 65 A of the Copyright Amendment Act 2012.

The fifth chapter “Contractual Restriction on Reverse Engineering” highlights the legal validity of contractual restriction on reverse engineering. For
that purpose, researcher discusses the emergence of software license and different forms of software license.

The sixth chapter “Role of Judiciary on Interface between Copyright and Competition Law” examines the interplay between copyright law and competition law in relation to reverse engineering of computer program. Critically analyze the doctrine of essential facility and copyright misuse as evolved by the judiciary.

The final chapter is on Conclusion and Suggestions. Findings are drawn on the basis of discussion made in previous chapters and important suggestions are made to preserve the reverse engineering of computer program for the growth of software industry.

*******