The Scope of Reverse Engineering of Computer Software under the Copyright (Amendment) Act, 1999: A Critique

K M Gopakumar
The Raoul Wallenberg Institute of Human Rights and Humanitarian Law
Lund University, Sweden.

(Received 7 September 2000)

This article attempts to examine the scope of reverse engineering of computer software under the recent amendment to the Copyright Act. The analysis is made in the background of a similar development in USA, Europe and India’s obligation under TRIPS Agreement. It contests the perception that the Act permits ample scope for reverse engineering and shows that the changes brought by the amendment give only a minimum space for reverse engineering. Though the amendment limits the space of reverse engineering, but it accommodates the market-oriented jurisprudence of copyright protection. The scope of reverse engineering under the Copyright Act even after amendment is the weakest in India. Limiting the scope of decompilation only for achieving interoperability exceeds India’s international obligation to protect the computer software. Therefore, limiting the scope of decompilation only for achieving interoperability is unwarranted.

The government has taken a series of steps to make the Indian intellectual property regime compatible with the TRIPS Agreement. The Copyright Act has been amended with the same purpose in mind, it also was an attempt to put the Indian copyright regime in the right perspective by melting down the stringent provisions related to fair use aspect of computer software in order to facilitate reverse engineering. India was one of the few countries, which extended copyright protection to computer programs in 1984. This was followed by another amendment in 1994, which enhanced the protection further by placing restrictions on the fair use provisions. The purpose of the recent amendment, i.e. the Copyright (Amendment) Act, 1999, is to provide legal recogni-
tion to reverse engineering of computer software by diluting the present provisions of the Act. This amendment is also noteworthy in the context that in other areas of intellectual property rights (IPR), India is trying to enhance the protection. This article attempts to examine the implications of the recent amendment on the scope of reverse engineering in the background of similar developments in USA and Europe. In doing so, the article contests the perceptions that the Act permits ample scope for reverse engineering and shows that the changes brought by the amendment give only a minimum space for reverse engineering. It also holds the view that the amendment has only accommodated market-oriented jurisprudence of copyright protection, i.e., to ensure the returns from investments, instead of protecting the expression of ideas.

For analytical purposes, the first part of study examines the need for reverse engineering. The second part discusses the scope of reverse engineering in the USA and Europe. The third part deals with the Indian position. Fourth part analyses international mandate on this issue and the conclusion.

**Reverse Engineering**

Reverse engineering is a recognized practice in all fields of technology resorted to analyse the advances made by new products. Reverse engineering, in the context of software, means the process of understanding functions of a program. Unlike other technologies, the software is mostly available to the public in an object code format. Therefore, it is not possible to decipher the functions of the program from the printout of an object code consisting of complex variations of binary numbers (combinations of 0 & 1). Over the years, programmers have developed different techniques to reverse engineer a program. They include reading program manuals to learn basic information regarding the programs’ operations, performing test runs by feeding a variety of inputs to examine the resulting outputs and converting the network to programs and then monitoring the program’s input and output while it runs. These techniques are commonly known as black box reverse engineering. However, these techniques do not provide all information about the interface specifications, which is a requisite for interoperability.

To achieve these, the programmers apply decompilation or disassembly techniques. These techniques involve translation of the program’s object code into a humanly readable source code form by using decompilation or disassembly programs. Essentially, disassembly and decompilation are one and the same. However, disassembly involves, specifically, a code translation from a machine-readable format to assembly language format. Decompilation, on the other hand, is the translation of the object code to higher-level language. But the decompiled version does not contain labels and comments, which may explain the program’s infrastructure because those elements will be stripped out of the program during compilation. Moreover, the program structure may also change from the original program. Thus, the term decompilation is a misnomer, though legally both 'disassemble' and 'decomposition' refer to the same thing.

Even though decompilation gives one a list of humanly readable instructions, it requires a high level of knowledge, understanding about the program, technical skill, expertise and intellectual contribution to identify and understand program function. Thus, it is an expensive and time-consuming process to try and decipher the interface specifications.
and protocols essential for compatibility. Hence, it is in many ways more difficult than writing a program right from scratch.

Reverse engineering is, however, required for the following purposes:

(i) to obtain interface specifications necessary for the development of an attaching product or to develop a competing product;

(ii) to obtain information about the capacity and performance characteristics of a program;

(iii) to debug and adopt the program for user's own environment;

(iv) a firm may reverse engineer its own program when it does not fully comprehend the operation of the software either due to the person who developed the program left the firm or due to lack of documentation.

Reverse engineering is opposed by the ultraprotectionists alleging that it results in piracy. The answer, however, depends on the definition of piracy. If one defines piracy as the unauthorized exact duplication of a product, then decompilation does not encourage piracy because for exact copying (piracy) one does not have to decompile the program. But, when the definition of piracy includes a competing program, this allegation is valid. However, this broader definition ignores the fact that the act of reverse engineering and the subsequent development of a competing program do not reproduce an exact copy of the program codes even though it may reproduce the result.

The decompilation right gives the programmer a right to analyse and study the program by resorting to intermediate copying. However, this right to intermediate copying is limited to discerning the idea behind the code and not to sell the copies of the program code. So, a literal copying of the program code still remains an infringement of copyright.

Software industry is characterized by the strong presence of network externality. As a result, the de facto standard products control the software market. The large corporations produce majority of de facto standard interfaces. Free access to interface information is necessary to make compatible and competing products and also for the maintenance of high performance standards. As a corollary to the reproduction right of interface specification, reverse engineering is to be made legal because, as stated earlier, it is necessary to understand the functions of the program. Further, reverse engineering is required on the ground of public policy to ensure competition and availability of products at affordable price to the consumer, dissemination of technology and also for the access to technology. Any prohibition on reverse engineering deprives the software industries in the developing countries an opportunity of developing compatible and competing products as well as access to the technology. Therefore, any copyright protection that goes beyond the program code to the structural elements of a program or its interfaces would impede technology transfer that is necessary for the development of domestic software industry in the developing countries. Further, any prohibition on reverse engineering is beyond the scope and purpose of the copyright jurisprudence. Computer software qualifies for copyright protection on the basis of its legal characterization as a literary work. The purpose of copyright protection of literary work is to protect the literary expression and not the idea behind it. Reverse engineering, irrespective of methods, is a practice to discern the ideas behind the program.
Therefore, using copyright protection to prohibit the decompilation would result in extending protection to the idea, which is beyond its purpose. In other words, copyright protection extends only to expressions and not to ideas.

US and European Scenario

The US Position

Except the reading of manuals all other reverse engineering techniques need to make an interim copy of the copyrighted program. In the case of decompilation, making of decompiled version also becomes an illegal copy. According to ultra-protectionists, this definitely is an infringement of copyright. There is no explicit provision under the US Copyright Act to permit reverse engineering. But two doctrines, namely Fair Use and Adaptation Rights are invoked to avail of the benefits of reverse engineering. According to the first doctrine, since object code is in a machine-readable form, one has to understand the idea of the program either from a source code or decompiled version. Thus, it constitutes a fair use of a copyrighted work. The second doctrine gives the owner of a copy the right to make a copy of the work and also adaptations for the user’s convenience. But most of the programs are distributed through licensing and not through sale. The legality of loading a program for the purpose of reverse engineering is held valid even in the presence of a provision contrary to this in the licensing agreement. This cleared the way for all methods of reverse engineering except that of decompilation. Appellate and Circuit Courts in Sega vs Accolade and Atari vs Nintendo decided the legal validity of the act of decompilation.

Sega vs Accolade

In Sega vs Accolade, the plaintiff made a computer game system comprising a console and a large number of game cartridges. Each cartridge contained an access code that was checked by the console before the game could operate. The defendant decompiled these lockout mechanisms and produced game cartridges, which were compatible to the console. Both cartridges contained a common piece of the code. The defendant added this to its program, so as to get access to the plaintiff's console.

The District Court held this as a copyright infringement. In the appeal, the Circuit Court reversed the earlier judgement and held that the decompilation of the object code was a fair use. In doing so, the court rejected Accolade’s arguments on the basis of Section 117 i.e. decompilation is a permitted use under Section 117 of Copyright Act. Also, copyright prohibits protection for ideas under Section 102.

The court based its reasoning on the fair use doctrine, under Section 107 of the Copyright Act. Under this section, to qualify as a fair use, the conditions are:

The purpose and character of the use including whether such use is of a commercial nature or is a non-profit, educational nature of the copyright, the amount and substantiality of the portion used in relation to the copyrighted work as a whole and the effect of use upon the potential market-value of the copyrighted work. Regarding the first condition, the court held that presumption of the unfair use could be reverted by the characteristics of particular commercial use. Regarding Accolade’s use, the court felt that “Accolade’s identification of the functional requirements for Genesis (i.e. the program) compatibility has led to an increase in the
number of independently designed videogame programs offered for use with the Genesis console. It is precisely this growth in creative expression based on the dissemination of other creative works and the unprotected ideas contained in those works, that the Copyright Act was intended to promote.\textsuperscript{26} Therefore, the use cannot be called an unfair use.

Regarding the nature of use, the court recognized software as utilitarian articles that contain many functional elements in addition to the protected elements. Further, the court said that even though “unprotected aspects of most functional works are readily accessible to the human eye... humans often cannot gain access to the unprotected idea and functional concept contained in object code without disassembling that code.”\textsuperscript{27} Finally on this issue the court held that “if disassembly of a copyrighted object code is \textit{per se} unfair use, the owner of the copyright gains a \textit{de facto} monopoly over the functional aspects of his work that were expressly denied copyright by the congress.”\textsuperscript{28}

About the third factor (amount of copying/substantiality), the court held that even though Accolade disassembled the entire program, but they used only the needed elements. Therefore, the court held that the amount of copying was minimal. The court finally remarked thus “where disassembly is the only way to gain access to the ideas and functional elements embedded in a copyrighted program and where there is a legitimate reason for seeking such access, disassembly is a fair use of copyrighted work as a matter of law.”\textsuperscript{29}

Atari vs Nintendo

The basic facts of this case are the same as that of the previous one. However, while obtaining the original program, Atari committed fraud with the copyright office. The federal court upheld the findings of the district court and held that Atari was not allowed to raise a fair use defence because it had obtained the copy of original program through unfair means. Nevertheless, on the issue of decompilation, the court said, “the Copyright Act permits an individual in rightful possession of a copy of a work to undertake necessary efforts to understand the work ideas, processes and \textit{modus operandi}.”\textsuperscript{30} However, the court put a limit to the fair use reproduction of a program. It must not exceed what is necessary to understand the unprotected elements of the work.

These cases established the rule that decompilation could be a fair use when it is the only way to gain access to the functional elements and if there is a legitimate reason for such an act.\textsuperscript{31} This means that the legality of decompilation depends upon the facts and the circumstances, which justify the fair use. This creates an uncertainty regarding the legal validity of decompilation. Moreover, it raises another question, as to whether anybody can have the right to decompile the original program if the owner of original program offers necessary information through licence. In such a situation the threat of litigation may be used as an effective weapon to discourage competitors from decompiling useful programs.

The ultra protectionists criticise these decisions because a decompilation, according to them, would undermine a copyright owner’s exclusive rights of reproduction, adaptation and translation of his or her work.\textsuperscript{32} This argument simply ignores the basic function of a copyright that it should protect only the expression and not the idea. Another criticism levelled against these decisions is that decompilation permits a second-comer to create a market substitute and reap the
benefits of successful programs after others have incurred the risk and expense of its development. This argument, again, presumes that a copyright's purpose is to guarantee the return of investments made for creating a work rather than protecting the expression of idea.

The new Digital Millennium Copyright Act provides an explicit exception to circumvention of technological protection measures of a lawfully obtained computer program in order to achieve interoperability. Hence this provision can be seen as a further legal recognition to the practice of reverse engineering.

**EU Position**

The European Council adopted the Directive on Legal Protection of Computer Programs in 1991, which explicitly permits reverse engineering. Articles 5, 6, and 9 of the Directive deal with the issues of reverse engineering. By virtue of Article 5, a person with a right to use the computer program is entitled to "observe, study, or test the functioning of the program in order to determine the ideas and principles which underlie any element of the program if he does so while performing any of the acts of loading, displaying, running, transmitting or storing the program" without the authorization of the right holder. This provision permits the lawful user of the program to reverse engineer the program using all the methods of black box reverse engineering.

The most controversial method of the reverse engineering, i.e. decompilation, is dealt with in the Article 6 of the Directive, which allows the lawful user to reproduce or translate the program which are "indispensable to obtain the information necessary to achieve the interoperability of an independently created computer program with other programs". However, this right is subjected to the following limitations. It should be performed by the licensee or by another person having a right to use a copy of a program or on their behalf by a person authorized to do so, the information necessary to achieve interoperability has not previously been readily available to the persons who were legally entitled to do so and the acts are confined to the parts of the original program which are necessary to achieve interoperability.

The main feature of this provision is said to be the limitation on the purpose of decompilation. Under the Directives, the decompilation can be done only to achieve interoperability between two programs. As stated earlier, one has to resort to decompilation not only to achieve interoperability but also to make competing programs, i.e. program with the same function or to check the security testing. Moreover, the wordings "to achieve interoperability of an independently created computer program with other programs," is raising concerns in many quarters because one needs decompilation not only to achieve interoperability between the programs but also between the programs and hardware. However, the preamble of the Directive states, "interconnection and interaction is required to permit all elements of software and hardware to work with other software and hardware." It is also pointed out that the translation and reproduction, i.e. decompilation of the program, is permissible under Article 5(1) under certain other circumstances, e.g. for debugging. Further, restrictions are placed on the use of information obtained from decompilation. The conditions are the information should not be used for goals other than to achieve interoperability of the independently created computer programs and it should be neither
given to others, except when necessary for
the interoperability of the independently cre-
ated computer program nor be used for the
development, production or marketing of a
computer program substantially similar in its expression, or for any other act which
infringes copyright. Further, Article 9
makes "any contractual provisions contrary
to Article 6 or to the exceptions provided for
in Article 5(2) ... null and void". Thus, the
Directive effectively checks the prohibition
of reverse engineering through licensing
agreements.

The scope of decompilation right available
after Sega and Atari decisions is considered
to be wider than the decompilation right
available in European Community under the
Directive on Legal Protection of Computer
Programs37. The Directive explicitly allows
decompilation by a licensee or another per-
son having a right to use a program to
achieve interoperability of an independently
created computer program with other pro-
gr ams38. Thus, the scope of decompilation
of a program is limited to interoperability39.
However, both Sega and Atari permit
decompilation, wherever necessary, to get ac-
cess to unprotected ideas if it is the only way
to get access and is coupled with a legitimate
reason40. In Atari, the Court explains the
word "legitimate reason" thus: "the Copy-
right Act permits an individual in rightful
possession of a copy of a work to undertake
necessary efforts to understand the work's
ideas, processes and methods of operation"
41. This wide scope for decompilation
may help the programmers to decompile the
program to get information about unpatent-
able algorithms in a program. However, a
further comparison of the Directive and the
US position reveals that, regarding the black
box reverse engineering methods, the Di-
rective is more liberal than the US law. Un-
der the US law, the right to resort to reverse
engineering short of decompilation derives
from the interpretation of Section 117 of the
Copyright Act, which is available only to the
owners of copyright and not to the licensees.
Contrary to this, Article 5(3) of the Directive
explicitly permits any lawful user to resort to
any method of black box reverse engineer-
ing techniques. The Directive also prohibits
any contractual restriction on reverse engi-
neering. This makes the European Directive
stronger than its US counterpart42. The lib-
eral scope of decompilation is neutralized by
the ambiguous position of licensing resur-
rection on reverse engineering.

Indian Position

India has been providing copyright protec-
tion for computer software since 1984. This
amendment among other things amended
the definition of literary work by including
computer program within its ambit. Apart
from this, the amendment did not bring any
change in the Copyright Act so far as the
computer software is concerned. This lac-
cuna was cured by another amendment in
1994, which provided extensive provision for
strengthening the protection of computer
software. The new amendment changed the
definition of literary work and inserted defi-
nitions of computer program, computer,
author, meaning of copyright and amend-
ment to fair use provision, known-use of in-
fringing copy and compulsory licensing.
This stringent norm, especially the amend-
ment of fair use provision, was provided for
ensuring protection to domestic computer
software industry. However, the urge to
amend the loopholes, the legal regime failed
to take into account realities of the domestic
software industry. Hence, the end product
was an Act that was not quite conducive to
the economic and marketing needs of the
software industry and it failed to achieve its objective of providing a viable environment in which it could flourish. It was this shortcoming that necessitated a further amendment in 1999. The 1999 amendment aims to dilute the stringent provision of 1994 amendment. Especially, the provision related to fair use by providing an explicit provision for reverse engineering.

Fair use exemption in copyright is for the purpose of safeguarding public interest. It permits the public to take certain parts of a work to use it in another work. Unlike the US Act, the Indian Act gives a list of exempted uses of a copyrighted work. Hence, there is no need for the courts to formulate any test to determine fair-use. The 1994 amendment took away the general fair-use defence with regard to computer software otherwise available under Section (52) (1) (a)43. The general fair-use defence permitted the free use of a work for 'private purposes, including research and criticism or review, whether of that work, or of any other work'. The denial of these two rights, namely, research and criticism, is inimical to public interest. These rights were denied on the ground that the general 'right to reproduce them for private use is not necessary in the interest of the bona fide users of computer programs'44. The provision per se prohibits the reproduction of a program by the purchaser even if it is for research and review. However, it gives the purchaser the right to make adaptations and back copies. The Act has curtailed the freedom of a researcher since he has to buy a copy for his research. Nevertheless, the impact of the Act would not be very drastic since exemption has been given to teachers, students and researchers in government and educational institutions45. However, private persons doing R&D would be affected by this provision. Thus, the Act would adversely affect individual software development.

The 1994 amendment added a new provision regarding fair dealing pertaining to computer software. It permits the lawful possessor of a software to make back copies or adaptations, but the use of these copies should be 'in order to utilize the computer program for the purpose for which it was supplied; or to make back copies purely as a temporary protection against loss, destruction or damage in order only to utilize the computer program for the purpose for which it was supplied'46. Thus, the Act limited the fair use exemption by putting two conditions. Quoting the above-mentioned conditions also restricts program adaptation. The first limitation, i.e. the utilization of the program only for the purpose for which it was supplied, is in effect a prohibition of all modes of reverse engineering except manual reading. This limitation further blocks the observation, study and test-run of a program so as to discern the ideas underlying the program. The hitch is that such a prohibition on software reverse engineering only helps the software giants to establish a legal monopoly in the market (in this case, the Indian software market). It denies the domestic industry any chance to familiarize itself with the latest technological advancements and develop indigenous programs and gadgets that are in tune with the global technology. What the legal system in India ignored is that such rights, when enjoyed by certain software giants in the US, Europe and Japan, were thoroughly rebuked by the international software fraternity. The denial of reverse engineering is also a denial of the legal recognition given to interoperability and compatibility of software. The provision also creates doubts as to whether the non-protection of interface specifications and
user interfaces will be properly adhered to. Therefore, it can be said that this provision went beyond the purpose of copyright protection by providing protection to the idea along with the expression (It has to be understood that this endeavour to protect the idea is the domain of patent protection and not that of copyright protection). By denying the right to reverse engineer, the 1994 amendment made a departure from the original purpose of copyright protection to market-oriented interpretation, i.e. to guarantee the return of investment made for creating the work instead of guaranteeing the accessibility.

The new amendment (1999) brought forth a few changes in the fair use provision pertaining to computer software. It added three new provisions in the Act in Section 52 (1) (aa). The new provisions read:

"(ab) doing of any act necessary to obtain information essential for operating interoperability of an independently created computer program with other programs by a lawful possessor of a computer program provided that such information is not otherwise readily available;

(ac) observation, study or test of functioning of the computer program in order to determine ideas and principles which underlie any elements of the program while performing such acts necessary for the functions for which the computer program was supplied;

(ad) making of copies or adaptation of the computer program from a legally obtained copy for non-commercial personal use".

Thus, the amendment permits decompilation or any other act required to achieve interoperability of an independently created computer program with other programs in the absence of ready availability of such information. Further, it explicitly permits other modes of reverse engineering by permitting observation, study or test of functioning of the computer program to determine the ideas and principles underlined in the program. But, this freedom is limited by the words “while performing such acts necessary for the functions for which the computer program was supplied”.

Making of a copy from a legally obtained copy for a non-commercial purpose is also permitted by the new amendment. Thus, these changes would definitely dilute the present legal position regarding reverse engineering, compatibility and research.

However, a close examination of these provisions reveals that the dilution of provisions is more apparent than real. The first problem arises with the explicit permission of ‘any act’ for achieving interoperability. The Bill uses the term ‘any act’ instead of ‘decompilation’ or ‘disassembly’. This may, at the outset, appear as a wider term, which includes not only decompilation and disassembly but also other modes of reverse engineering mentioned in Section 52 (1) (ac). However, the problem lies in the actual scope of such a provision. The new Amendment permits these ‘acts’ only for achieving interoperability. The Bill uses the term ‘any act’ instead of ‘decompilation’ or ‘disassembly’. This may, at the outset, appear as a wider term, which includes not only decompilation and disassembly but also other modes of reverse engineering mentioned in Section 52 (1) (ac). However, the problem lies in the actual scope of such a provision. The new Amendment permits these ‘acts’ only for achieving interoperability. Thus, the scope of decompilation is restricted to achieving interoperability. This is akin to the provision given in the EC Directive. Decompilation is not only required for interoperability but also for making a competing program. Further, it is also needed for purposes of security testing. Again, the scope of decompilation is restricted by permitting it only “for operating interoperability of an independently created program with other programs”. One has to achieve interoperability not only with the software but also with the hardware. In the US, decompilation is not only required for interoperability but also for making a competing program. Further, it is also needed for purposes of security testing. Again, the scope of decompilation is restricted by permitting it only “for operating interoperability of an independently created program with other programs”. One has to achieve interoperability not only with the software but also with the hardware.
pilation right is available to discern the idea of program if that is the only way to understand the idea of the program. There is no valid reason to restrict the scope of a provision, especially when it can positively help the technological development of the domestic hard and software industries. Further, according to the new provision, reverse engineering can be resorted to only in a situation wherein such information is not otherwise readily available. Thus, the provision presupposes the party to first approach the owner of the copyright for the information required for interoperability before opting for reverse engineering. In the absence of any clear definition, information subject to licence fee may be interpreted as within the scope of 'information readily available'. In that case, one has to pay royalty to get the information about interfaces. This would mean that the provision, which permits interoperability, would become nothing but a showpiece.

Further, the word "any act" in Section 52 (ab) creates confusion regarding the different modes of reverse engineering except decompilation and disassembly. The word "any act" encompasses all forms of reverse engineering. Does it mean that other forms of reverse engineering other than decompilation and disassembly constitute a violation of fair use if it is put to use for any other purpose than interoperability? Even though Section 52 (ac) explicitly permits other modes of reverse engineering "while performing such acts necessary for the functions for which the computer program was supplied", it does not clarify the position. Moreover, it adds to the confusion since the above provision can be interpreted to mean as prohibiting use of those methods exclusively to discern ideas behind a program. Lastly, confusion persists regarding the information derived from performing such acts under Section 52 (ac), as to whether it can be used for interoperability even in the presence of readily available information. The Act does not provide any clue on these issues. One has to wait for opinion from the courts for clarification.

After the Sega and Atari cases, the US law on decompilation did not impose any purpose-oriented limitation. The only limitation was that it was possible to resort to decompilation only in the absence of any other legitimate way. The European position regarding reverse engineering, except decompilation, is very clear in Article 5(3) EC Directive on Legal Protection for Computer Program. Decompilation is permissible only to achieve interoperability in the absence of ready availability of such information. But, the strongest point in the European provision is that it prohibits any contractual restrictions on any mode of reverse engineering.

In India, though we follow a close analogue of the European Directive, we have failed to provide any provision against contractual restrictions. The envisaged invalidity of the shrink-wrap agreement is not totally ensured. Moreover, other contractual restrictions can easily prevent the purchaser from reverse engineering. This makes the Indian reverse engineering provision the weakest amongst the three.

**International Obligations**

The International initiative to protect computer program was started by WIPO in the late 70s. WIPO's focus then was to bring about a *sui generis* mode of protection to computer software. However, after the US decided to extend copyright protection to computer software, proposal for a *sui generis*
protection was shifted to copyright. The WIPO initiative again gained momentum in 1989 when attempts were made to include a protocol to Bern Convention without engaging in a full revision of the Convention. This process culminated in the adoption of WIPO Copyright Treaty in 1996. Around the same time, the Uruguay round of trade negotiations took up Trade-Related Aspects of Intellectual Property Rights (TRIPS) and concluded TRIPS Agreement in 1994. The protection of computer software found place in the TRIPS Agreement as well as in the WIPO Copyright Treaty. The WIPO Copyright Treaty adopted provisions, which were in line with TRIPS Agreement. Thus, the issue of protection of computer software got explicit mention in these two International Treaties. Even though there is no provision in the Bern Convention relating to computer software protection, TRIPS and WIPO Copyright Treaty ipso facto makes signatories of these treaties to adhere to the rules of Bern Convention. The following discussion would focus the question whether there is an international obligation against reverse engineering of computer software in the TRIPS and WIPO Copyright Treaty.

Article 10 of TRIPS Agreement obligates contracting parties to protect “computer programs whether in source or object code, ... as literal work under the Bern Convention”. One should read this in the light of Article 9 (2) of TRIPS Agreement, which reads: “copyright protection shall extend to express, and not to ideas, procedures, methods of operations of mathematical concepts as such”. WIPO Copyright Treaty also explicitly provides protection for computer software. It requires the contracting parties to protect the computer software as a literary work within the meaning of Article 2 of the Bern Convention irrespective of mode or form of their expressions. Further, the protection, according to Article 4, extends only on expression and not to ideas, procedures, methods or operations and mathematical concepts as such.

As stated earlier, protection of computer software under TRIPS and Copyright Treaty never went beyond expression. The only obligation under these treaties was to ensure the protection of a program as a whole. Further, the treaties permit the signatories to limit the exclusive right of author of the protected work. In the absence of any obligation to protect the underlying technical ideas encompassed in software, the program can be reverse engineered. Moreover, there is no limitation regarding the scope of reverse engineering. That is to say, even though wholesale copying of a computer program is prohibited, “the practice of re-implementing the functional components of a protected program in clones is not prohibited”. Thus, an independently developed program, which performs the same function as that of the earlier program, does not infringe the copyright. In other words, one can reverse engineer a program for any purpose including developing competing program. This is what Singapore and Australia did by amending their respective copyright acts to permit reverse engineering even to make competing programs. Ignorance of this legal interpretation again puts the Indian Copyright Act squarely within the market-friendly jurisprudence.

Conclusion

Reverse engineering is required to provide access to technology and introduce or encourage competition in the software industry especially for the developing countries. India, as a potential software superpower, should, therefore, have a soft regime on re-
verse engineering, especially on decompilation, in order to increase the access and dissemination of technology as well as to ensure software products at an affordable price. The above discussion shows that the Indian regime is the weakest among the three legal systems examined here. The drafters of the Bill totally ignored the international mandate and the manoeuvring space available to India. As a party to TRIPS, India is bound to provide only minimum protection prescribed by the Agreement. The only obligation under TRIPS pertaining to computer software is to extend protection against literal copying and it does not extend to protecting functional elements of the software.

Limiting the reverse engineering process especially the decompilation only for achieving interoperability is a narrow interpretation of the treaty provisions. The correct perspective would be that the protection is against wholesale copying. Broad scope for reverse engineering would promote competition and also help in the expansion of opportunities of firms in developing countries to develop substitutes or interoperable programs and to enhance their participation in the world market. Therefore, national legislation can provide all methods of reverse engineering, including decompilation/disassembly as legitimate practices, which will help develop interoperable as well as competing programs. Limiting reverse engineering especially decompilation only for achieving interoperability is, therefore, unwarranted as it retards competition as well as dissemination of technology within the domestic industry. It is also required to put the copyright in its right perspective. The present Indian Act is to be amended so as to do away with this over protective norm.

Acknowledgement
The author wishes to thank Mr K Balakrishnan (CUSAT) and Mr Chery Jacob (JNU) for editorial assistance.

References
3 Ibid.
4 Supra note 1 p15.
6 Supra note 1 p17.
7 Ibid.
8 Ibid.
9 Ibid.
12 Ibid.
13 Lorrie Ackerman, Should Reverse Engineering of Computer Software Through Intermediate Copying be
Prohibited." Available at http://www.eff.org/pub/intellectual_property/reverse_engineering.article


17 Karjala Dennis S, Theoretical Foundation for the Protection of Computer Program in Developing Countries, paper presented at International Conference on Intellectual Property Rights in Computer and their Impact on Developing Countries, 19-21 August 1993, Banglore, p16

18 Art.9(2) of TRIPS

19 *Supra* note 1, p 17

20 The Fair Use Doctrine is a Valid Defence to an Otherwise Valid Act of Copyright Infringement. According to Section 10 the limitations on exclusive rights or fair use are spelt out as follows:

"Notwithstanding the provisions of Sections 106 and 106A, the fair use of a copyrighted work, including such use by reproduction in copies or phonorecords or by any other means specified by that section, for purposes such as criticism, comment, news reporting, teaching (including multiple copies for classroom use), scholarship or research, is not an infringement of copyright. In determining whether the use made of a work in any particular case is a fair use. The factors to be considered shall include -

(1) the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;

(2) the nature of the copyrighted work;

(3) the amount and实质性 of the portion used in relation to the copyrighted work as a whole; and

(4) the effect of the use upon the potential market or value of the copyrighted work.

The fact that a work is unpublished shall not itself bar a finding of fair use if such finding is made upon consideration of all the above factors".


21 17 USC 117, According to this Section the owner of the program can make backup copy of the program. It reads: "(a) Making of Additional Copy or Adaptation by Owner of Copy. - Notwithstanding the provisions of Section 106, it is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided: (1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or (2) that such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful".

22 Valut vs Quaid847 F. 2d. 255 (5th CIR), 1988
Accolade's Section (102) based argument consisted of following steps:

1. Computer programs contain ideas unprotected by copyright.
2. If a program developer distributed it in object code format, a competitor often can, discern the program's unprotected ideas only by disassembling the program.
3. If such disassembly constituted copyright infringement then copyright in effect would protect the program's ideas.
4. Because Section 102B prohibits any direct copyright protection for ideas, it must also prohibit copyright from being used indirectly to protect ideas i.e. by restricting disassembly.

Sega vs Accolade, 977 F.2d, 1523, quoted by Band and Katoh, Supra note 1, p204.

Id. at 1525, p 204.

Id. at 1526, p 206.

Id. at 1526, p 208.

Atari vs Nintendo, 975 F.2d at 842 Id. p213.

Id p214.

Supra note 10


17 USC § 1201(0)

Article. 6(1)

Article 6(2)


There is a view that EU Directive does permit decompilation for the purpose of creating competing programs. See www.interop.org/USTR

Supra note 1, p206.

Supra note 1, p213.

In the US there are legal provisions against the restrictions on the reverse engineering based on the copyright and patent pre-emption of such restrictions.

Section 52 (1) (a), (1) The following acts shall not constitute an infringement of copyright, namely:

a fair dealing with a literary, dramatic, musical or artistic work not being a computer program for the purpose of-(i) private use, including research;
(ii) criticism or review, whether of that work or of any other work.

The Copyright (Second Amendment) Bill, 1994. Section 52 (1) (a) and notes on clause 17, p 22.

Ibid.

Section 52 (1) (aa).
47 Section 7, The Copyright (Amendment) Bill 1999.

48 Section 52 (1) (ab).

49 Article 9(2) and 10 of the Berne Convention, Article 10 of TRIPS, Article 13, Article 10 of Copyright Convention.


51 UNCTAD, The TRIPS and Developing Countries, UN, New York, 1996, p40.


53 Section 52 (1) (ab) of the Indian Copyright Act (as amended now) reads: “The doing of any act necessary to obtain information essential for operating interoperability of an independently created computer program with other programs by a lawful possessor of a computer program provided that such information is not otherwise readability available....” Article 6 (1) of EC Directive limits Decompilation rights only for interoperability it reads: The authorization of the right holder shall not be required where reproduction of the code and translation of its form within the meaning of Article 4 (a) and (b) are indispensable to obtain the information necessary to achieve the interoperability of an independently created computer program with other programs....

54 Supra note 51, p39.