Annual ADFSL Conference on Digital Forensics, Security and Law

2011 Proceedings

May 25th, 2:00 PM

Software Piracy Forensics: Impact and Implications of Post-Piracy Modifications

Vinod Bhattathiripad
Cyber Forensic Consultant, Polpaya Mana, Thiruthiyad, Calicut, Kerala India, vinodpolpaya@gmail.com

S. Santhosh Baboo
Reader, P G & Research, Dept of Computer Science, D.G.Vasihanv College, Chennai, India, santhos2001@sify.com

SOFTWARE PIRACY FORENSICS: IMPACT AND IMPLICATIONS OF POST-PIRACY MODIFICATIONS

P. Vinod Bhattathiripad
Cyber Forensic Consultant
Polpaya Mana
Thiruthiyad, Calicut-673004
Kerala, India
Telephone: +91-495-2720522, +91-94470-60066 (m)
E-mail: vinodpolpaya@gmail.com; vinodpolpaya@yahoo.co.in

Lt. Dr. S. Santhosh Baboo
Reader
P G & Research
Dept of Computer Science
D.G.Vasihanv College
Chennai, India.
E-mail : santhos2001@sify.com

ABSTRACT

Piracy is potentially possible at any stage of the lifetime of the software. In a post-piracy situation, however, the growth of the respective versions of the software (both the original and pirated) is expected to be in different directions as a result of expectedly different implementation strategies. This paper shows how such post-piracy modifications are of special interest to a cyber crime expert investigating software piracy and suggests that the present software piracy forensic (or software copyright infringement investigation) approaches require amendments to take in such modifications. For this purpose, the paper also presents a format that is jargon-free, so as to present the findings in a more intelligible form to the judicial authorities.

Keywords: Piracy, post-piracy modifications, software piracy, source code, copyright, software copyright infringement, software piracy forensics, database forensics, MIS forensics, AFC, SCAP, technical expert, substantial similarity test, CDAC

1. INTRODUCTION

Piracy is potentially possible at any stage in the lifetime of the software. If and when that happens, the original and pirated versions of the software will continue to be used contemporaneously. This being so, in the post-piracy period, the profile of the pirated[1] could well be in a different pattern to that of the original[2] as both the original developer as well as the pirate may modify the respective versions in their own ways. Because of this, although the original and the pirated software are prone to grow functionally in almost the same direction (because the post-piracy life time of the pirated software is in the same functional area of expertise as that of the original), the growth is expected to be with different modification strategies (because both are handled by different persons). This phenomenon of different patterns of growth is a very valuable and useful dimension of study for the expert in cyber forensics. A

[1] Throughout this article, pirated means the allegedly pirated software

[2] Throughout this article, original means the version of the software that the complainant submits to the law enforcement agency for software piracy forensics. This article 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.

proper study of post-piracy modification of the pirated will contribute substantially to the reliability of software piracy forensic investigation. This article attempts to discuss the impact and implications of post-piracy modifications in software piracy forensics (or software copyright infringement investigation) and to suggest that proper amendments be made in the existing forensic approaches / techniques so that evidence concerning post-piracy modifications gets proper consideration and treatment.

Software piracy forensic investigation often requires comparison of the original with the pirated by juxtaposing the two. In order to perform the task of comparing two software packages, several software tools are used and these tools are based mostly on academically accepted mathematical techniques and theoretical frameworks like Discourse Analysis (Van der Ejik, 1994), SMAT (Yamamoto et al, 2004), and MOSS (Lancaster and Culwin, 2004). A recently (November-2009) edited work "Handbook of Research on Computational Forensics, Digital Crime, and Investigation: Methods and Solutions" (Chang-Tsun Li, 2010) prescribes SCAP (Frantzeskou, 2007) for comparison of two software packages. An exception to all these (because of the judicial acceptance in the US) is the theoretical frame work called AFC (Abstraction-Filtration-Comparison) (Walker, 1996) which has been professionally implemented by a French software firm European Software Analysis Laboratory in the form of the product namely SIMILE Workshop (ESALab, 2007). All these techniques, theoretical frameworks and tools are capable in their respective areas of software piracy investigation. Even so, none of them properly and adequately deal with matters related to post-piracy modifications in the pirated and all these need to be made sensitive to the implications of post-piracy modifications.

Post piracy modifications need also to be incorporated into many other theoretical proposals and studies in software piracy forensics. For instance, the Ginger Myles (2006, p.69), proposes watermark (as a weaker evidence) to indicate that "one program is likely to be a copy of the other", needs further explanation on extending watermarking to post-piracy modifications in the pirated. Anthony Reyes (2007) beautifully discusses areas of difficulties, misconceptions and flaws in the cyber investigative methodology by explaining techniques for preparing for prosecution, testifying, and incidence response, solving legal issues, conducting seizure procedure, performing data analysis, and preventing of cyber crimes, including software piracy, but this work requires further considerations on analysis of post-piracy schematic changes. Another example is the work on software forensics by Robert M. Slade (2004), where he explains from his experience, several ways of collecting evidence of software piracy and presents overviews of forensics programming, plagiarism detection, code analysis, source code recovery and even forensics linguistics. Even so, the book does not cover matters related to post-piracy modifications in the pirated.

2. ESTABLISHING THE CRIME

As mentioned above, the starting point of the investigation into piracy is the juxtaposed comparison of the original and the alleged pirated versions. This delicate and demanding situation of comparing two software packages arises usually when one party lodges a complaint of software piracy or copyright infringement against the other. A full-fledged forensic investigation of the pirated software has to be done to establish piracy. As software piracy investigation involves technical comparison, the judge usually appoints an uninvolved cyber forensic expert for the task. Given the source codes from two different software systems, the technical expert concentrates on digging out the pieces of potential evidence of copyright infringement by evaluating the similarities and commonalities that form the basis for validating or invalidating the alleged crime. The duty of the cyber forensic expert is to establish possible piracy through a rigorous formulation of *statistical occurrences* of the data structures, variables, data base tables, fields, modules, procedures, logic, remark, error and blunders in the allegedly pirated software and arrive at several values, preferably in percentages, to indicate the strength of piracy (Author, 2009, p.54), all of which require comparing the original and the pirated source codes, database schemas and procedures. Alternatively, the cyber forensic expert can abstract the original as well as pirated, filter out globally common elements from them and then compare the

remaining two kernels in order to establish copyright infringement (Walker, 1996). Either way, the procedure needs to take into account evidence concerning post-piracy modifications.

3. PROCEDURE

As a prelude to comparing the two software systems, the cyber forensic expert can ask the original developer (the complainant) to make available their pre-modified version of the source code, the embedded images and finger prints, the database procedures and the database schemas that were prevailing at the time of piracy (see footnote 2 above). At the same time, the source code, the embedded images and finger prints, the database procedures and the database schemas of the pirated are generally made available for comparison by the police or judiciary through a seizure procedure (Authors, 2009, p.177) or through a disclosure procedure wherein the technical expert (or sometimes the plaintiff too) has direct access to the defendant's source code (Hollaar, 2002, p103). (This procedure might vary from country to country). It is highly unlikely that this (or thus made available) version is the pre-modified version of the pirated software. The seizure procedure usually ends up seizing some as-and-when available version, mostly a modified / customized version of the pirated software, leaving the cyber forensic expert with this version of the pirated software to compare with the original.

4. THE IDENTIFICATION OF POST-PIRACY MODIFICATIONS

The modified / customized version of the pirated software that is made available through the seizure would most certainly have gone through a few, if not quite a lot of, modifications. This being so, before listing out the similarities (and commonalities) and making an expert judgment from their statistical representation and profile, the cyber forensic expert has to first generate, ideally, the originally pirated pre-modified version of the software out of the seized version by identifying and filtering out the post-piracy modifications, if any, from it. These post-piracy modifications can be found in various parts of the pirated software, namely, the source code, object files, embedded fingerprints and images, database procedures, and/or the database schemas and so, the cyber forensic expert has to necessarily identify and filter out all identifiable post-piracy modifications from all these parts, one by one. The possibility of their tainting the statistical rigor of the results of the comparison will thus be eliminated or at least minimized. The expert has to first convert the seized, pirated into its pre-modified infant form. The objective of and hence the emphasis on this process is not the detection or confirmation of piracy but the identification and filtering out of all post-piracy modifications by a more rigorous scrutiny of differences between the two codes. The output of this initial process will be something closest to the pre-modified version, which forms the basis for a reliable eventual comparison with the original.

5. POST-PIRACY MODIFICATIONS – THE WHY AND THE HOW

The above facts demand a detailed study on post-piracy modifications and ways to incorporate their role, effect and impact in the report to the court. Different techniques to analyze post-piracy modifications are required because post piracy changes can happen along a variety of parameters, in a variety of ways, and for a variety of reasons not all of which may be visible, noticeable and reliable initially during the cyber forensic investigation. However, on detailed investigation, they all can be seen to become relevant and can largely influence the cyber forensic report.

Difference is not exculpation from piracy: While similarity between two sets of software is most certainly indicative of piracy, difference is not exculpation from piracy either. In fact it is the differences that trigger the need for careful observation since they may be the result of post-piracy modifications. Such modifications may be motivated by a number of factors. For instance, one motivation for modification could be a customer demanding an additional feature in the software.

Another could be a government-directive to be incorporated in business. In both cases the software will have to be subsequently modified accordingly. In order to incorporate a customer request or government directive into the software, the pirate may modify, say, the structure of the table by introducing one or more new fields into it. While a government directive can bring about a modification in both original and the pirated (with different implementation patterns), the implementation of a customer request by the pirate brings about a change only in the pirated. Such a modification would induce difference between the pirated and the original data base tables and it is the duty of the cyber forensic expert to consider and properly question these differences during software piracy investigation. In addition to customer requirements and government directives, the pirate himself / herself may introduce intentional changes in the database schema in order to escape copyright violation litigations in the future and such intentional changes also cause questionable differences. It needs to be stressed that customer requests, government directives, and intentional changes are only some of the potential reasons and motivations that can cause questionable difference in the pirated from the original, and database is only one of the areas where such questionable differences can be found in the pirated.

Differences exist in various forms: During the post-piracy lifetime of the software, the pirate might modify database schema (which formally defines the tables in each database, the fields in each table, and the relationships between fields and tables), by adding, removing or editing a few fields, as part of either the post implementation tuning up or of the customization of the pirated software. Thus, any difference found in the schema of the pirated can either be in the form of the presence of one or more additional fields (that are absent in the original) or absence of one or more fields (that are found in the original) or modifications in names and/or other properties of any of the fields already existing intact in both original as well as pirated.

An example of suspected post-piracy modifications: These forms of post-piracy modifications and the resulting forensic challenges / difficulties[3] can be better explained with an example of a database-related situation. Table-1, which was extracted from a software comparison report (Author, 2002, p.14), gives a sample of database table-level comparison. The first part of the table corresponds to the original and the second, to the pirated. Fifteen out of sixteen fields in the original are found exactly in the same sequence in the pirated also (93% similarity or 93 % of the fields in the original can be mapped to at least one field in the pirated) and 15 out of 21 fields in the pirated are found in original also exactly in the same sequence (71% similarity). While these two percentages are fair enough to give some clue to possible piracy, there is still scope for the expert to further analyze the two database schemas with the intention of improving on the above two percentages, suggestive of suspected piracy. A further analysis of the remaining 1 field in the original and 6 fields in the pirated results in more reliable percentages (of nomenclature level piracy) than the above two. This is better explained here with the one field namely USERNAME in the original which can be seen to have some correspondence with two fields, namely, CREATEDUSER and MODIFIEDUSER in the pirated. Just as the field name USERNAME is one of the globally used variable names to save the name of the author of the transaction, the two field names, viz., CREATEDUSER and MODIFIEDUSER in the pirated are also globally used to save the names of the authors of the transaction and thus, a correspondence can be attributed between them. While the field, USERNAME, has some degree of nomenclature level similarity with CREATEDUSER and MODIFIEDUSER, the degree of dissimilarity can be possibly because of a post-piracy development, in which the 'pirate' himself may have replaced USERNAME with two fields, namely, MODIFIEDUSER and CREATEDUSER. This

[3] These challenges / difficulties are usually explained using theoretical situations involving source codes but the situations used in this article are live and data base related. Live situations are often more valuable than theoretical ones. Moreover, any post-piracy modification in the database would generally subsume the corresponding change in the respective source code too.

possibility is further strengthened by the similarity[4] in the other properties of these fields. For instance, all these three fields are of type CHAR, and are of length six (see Table-1). Thus, this strong possibility of post-piracy modification demands either re-calculation of the above two percentages or incorporating this possibility separately in the cyber forensic report. By mapping USERNAME in the original to both the CREATEDUSER and MODIFIEDUSER in the pirated, it can be seen that each of the 16 fields in the original can be mapped to at least one field in the pirated, and thus the above given 93% similarity in effect becomes 100%. Similarly, by mapping both the CREATEDUSER and MODIFIEDUSER in the pirated to USERNAME in the original, it can be seen that 17 out of 21 fields in the pirated can be mapped to at least one field in the original, which increases the above given 71% to 81%. As 17 out of 21 fields in the pirated could successfully be mapped to at least one field in the original, the remaining 4 fields, namely ACCTRANSCHEQUEDATE, ACCTRANSISSUEBANK, INTERNALENTRY, and VOUCHERTYPE, also require further attention and analysis, as these 4 fields also can possibly be post-piracy fields. Thus, an analysis of the modifications happened in the schemas of the pirated does shed further light on the suspected piracy. This example illustrates how a post-piracy modification can happen along database fields (one of the above listed parameters of a database) and shows a way for the expert to overcome the resulting forensic challenges / difficulties by using his / her expertise, intuition and common sense in identifying post-piracy modifications. The test used in the above example is “Substantial similarity “ test (Davis, 1992).

[4] Further, such strong similarity can also happen if the software was originally made by the respondent but later pirated and then unethically copyrighted by the complainant. The law enforcement agency needs to collect evidence from the respondent and from other sources and prepare the case accordingly.

Table -1: Comparison of the structures of two database tables (Author, 2002, p.14)

	Original software's database table structure
TABLE NAME: ACCOUNTTRANSACTIONS	
Field names and properties	
1	ACCOUNTHEAD CHAR(8) NOT NULL
2	FINYEAR CHAR(4) NOT NULL
3	ACCTRANSVOUCHERNUMBER CHAR(8) NOT NULL
4	ACCTRANSBILLNUMBER CHAR(11)
5	ACCTRANSCHEQUENUMBER CHAR(10)
6	ACCTRANSCREDIT NUMERIC(12,2) DEFAULT 0.00
7	ACCTRANSDATE DATE
8	ACCTRANSDEBIT NUMERIC(12,2) DEFAULT 0.00
9	ACCTRANSDESCRIPTION CHAR(300)
10	ACCTRANSRECDATE DATE
11	ACCTRANSRECONCILE CHAR(1) DEFAULT 'N'
12	ACCTRANSTYPE CHAR(2)
13	COSTCENTRE CHAR(2)
14	DIVISION CHAR(2)
15	USERNAME CHAR(6) NOT NULL
16	MACHINEID CHAR(10) NOT NULL
	Seized (allegedly pirated) software's database table structure
TABLE NAME: ACCOUNTTRANSACTIONS	
Field names and properties	
1	ACCOUNTHEAD ACCOUNTHEAD_DM /*CHAR(8) CHARACTER SET NONE*/ NOT NULL
2	FINYEAR /*RDB$914*/ CHAR(4) CHARACTER SET NONE NOT NULL
3	ACCTRANSVOUCHERNUMBER /*RDB$915*/ CHAR(8) CHARACTER SET NONE NOT NULL
4	ACCTRANSBILLNUMBER /*RDB$916 */ CHAR (11) CHARACTER SET NONE
5	ACCTRANSCHEQUENUMBER /*RDB$917 */ CHAR(10) CHARACTER SET NONE
6	ACCTRANSCREDIT AMOUNT_DM /*NUMERIC(15,2 */ DEFAULT 0.00
7	ACCTRANSDATE /* RDB$918 */ DATE
8	ACCTRANSDEBIT AMOUNT_DM /*NUMERIC(15,2) */ DEFAULT 0.00
9	ACCTRANSDESCRIPTION /*RDB$919 */ CHAR(300) CHARACTER SET NONE
10	ACCTRANSCHEQUEDATE /*RDB$920 */ DATE
11	ACCTRANSISSUEBANK NAME_DM /*CHAR(50) CHARACTER SET NONE*/
12	ACCTRANSRECDATE /*RDB$921 */ DATE
13	ACCTRANSRECONCILE BVALUE_DM /*CHAR(1) CHARACTER SET NONE */ DEFAULT 'N'
14	ACCTRANSTYPE BOOKCODE_DM /*CHAR(2) CHARACTER SET NONE */
15	COSTCENTRE /*RDB$922*/ CHAR(2) CHARACTER SET NONE
16	DIVISION /*RDB$923*/ CHAR(2) CHARACTER SET NONE
17	INTERNALENTRY BVALUE_DM /*CHAR(1) CHARACTER SET NONE */ DEFAULT 'N'
18	VOUCHERTYPE /*RDB$924*/ CHAR(1) CHARACTER SET NONE*/ DEFAULT 'E'
19	MODIFIEDUSER USERID_DM /*CHAR(6) CHARACTER SET NONE */
20	CREATEDUSER USERID_DM /*CHAR(6) CHARACTER SET NONE */ NOT NULL
21	MACHINEID MACHINEID_DM /*CHAR(10) CHARACTER SET NONE */

Post-piracy modifications and AFC test: If the above investigation is done using the AFC test (the recognized test in the US judiciary for software copyright infringement investigation), most of the data base fields mentioned above can be filtered out (from the original as well as pirated) during the filtration stage of test (Hollaar, 2002, p89) and in such case these fields will not be available for final comparison. This is because, most of these data base fields carry names which are globally not uncommon and thus these fields may be treated under "widely accepted programming practices within the computer industry". This sort of filtration of AFC can seriously impair the evaluation of the evidence concerning post-piracy modifications in the pirated and so defeat the purpose of the software copyright infringement investigation. Thus, this sort of filtration is tantamount to an act of discarding valuable digital evidence of post-piracy modifications in the databases. (What is required here is to re-design the filtration stage of AFC so as to avoid filtering out the possible evidence of post-piracy modifications from the pirated.)

Factors encouraging post-piracy modifications: Two factors that encourage the possibility and extent of post-piracy modifications are; (a) the time lapse between the actual act of piracy and the complaint from the original developer; and (b) the market of the pirated version. It is commonsense to believe that there is a direct though not systematic connection between the extent of post-piracy modifications and the time available to do it on the one hand and the nature and extent of the consumer (of the pirated version) on the other. A good illustration of the importance of the time factor is the suit Sesame Software Solutions Vs. Perfect Software, filed in 2007, the final verdict of which is still pending in a court in India. The complainant in the case had alleged that four of his former employees had appropriated his software product and were marketing it as their own since the time they left his employment six years earlier. The alleged software seized on court order through a raid (Author, 2007) was the latest (as-is where-is) version and very probably a modified version. The court appointed the cyber forensic division of Centre for Development of Advanced Computing (CDAC, Thiruvananthapuram, India) as the expert to investigate the piracy. In a case like this, since there is every possibility that the software would have been modified drastically during the six years, no attempt by the CDAC to compare the original with the pirated would yield realistic results if the post-piracy modifications are ignored, particularly in view of the long period of six years involved. The incidence of such suits may well be on the rise globally.

Choice of Tool matters: In the situation of software piracy forensics, often what determines the credibility of the result is not just a matter of the professional status of the expert, or the dexterity of the analysis but also of the appropriate choice of tools and approaches used in the comparison of software. For instance, the file comparison utility software used by CDAC in the above case was not versatile enough to identify and filter out the post-piracy modifications. The weakness arising from non-use of the required tool is illustrated in Table 2, extracted from pages 71 and 119 of the cyber forensic analysis report performed by CDAC (2008) in the above mentioned suit, Sesame Software Solutions Vs. Perfect Software. This table provides an instance of table level comparison, in which the file comparison utility software (whose name is not mentioned in the report) used by CDAC found that some of the data base fields in the pirated do not prima facie appear to be fully similar to those in the original (see the last part of table-2). Nowhere in table-2 (and also in the report) is there any mention about the possible post-piracy modifications that could have happened during the six-years of post-piracy life of the pirated. This laxity can be because of the lack of skills of the file comparison utility used in this case by CDAC. A supplementing and in-depth manual comparison (with the intention of identifying the post-piracy modifications) would have easily revealed that the fields AcgCode and ACGrCode differ only by the character 'r', AcsCode and AcSuCode, by the 'u' and AcsName and AcSuName by the 'u'. In other words, the first three fields in the pirated are different from the respective three fields of the original only by a single character each and this difference can possibly be a result of post-piracy modifications with an explicit intention of obfuscating similarities. This possibility has not been explained properly in the report. The cyber forensic expert could have

'properly' reported to the court that the three "relatively similar fields" (see Table-2) differ only by a single character and that this minor difference (favouring the alleged culprit) could possibly have been brought about by an intentional act of obfuscation. In other words, a thorough expert would question and further explore manually the minor, nominal degree of dissimilarity in the three "relatively similar fields" to weed out the possibility of a deliberate act of obfuscation and judiciously report the findings to the court. Once the questionable nature of such factors has been pre-supposed, that would then logically form a legitimate precedent for a similar investigation of the remaining two fields namely UsrCode and UsrEnteredOn, which too may well be suspected as a post-piracy add-on. Almost all other table level comparison results in this report (CDAC, 2008) are incomplete in this manner. Quite a lot of such superficial differences, thus, need further manual supplementary analysis to establish their legitimacy and such manual analysis draws upon clear insight, commonsense, hands-on experience, and intuitive skill of the expert.

Table-2: Results of comparison of the two database table structures (CDAC, 2008, p.71, p.119)

Original software's database table structure	
TABLE NAME: AcSubGroup	
Field names and properties	
1	[ACGrCode] [int] NULL
2	[AcSuCode] [int] NULL
3	[AcSuName] [varchar] (30) NULL
Seized (allegedly pirated) software's database table structure	
TABLE NAME: AcSubGroup	
Field names and properties	
1	[AcgCode] [tinyint] NOT NULL
2	[AcsCode] [tinyint] NOT NULL
3	[AcsName] [varchar] (40) NOT NULL
4	[UsrCode] [varchar] (5) NOT NULL
5	[UsrEnteredOn] [datetime] NOT NULL
Results of comparison of the above two table	
	Number of fields in the allegedly pirated: 5
	Number of fields in the original : 3
	Same fields : 0
	Relatively similar fields : 3
	Not similar fields : 2 fields in the allegedly pirated and 0 fields in the original

A further point to look into while choosing the right tool is the tool's ability to analyze the positioning or placement of the post-piracy add-ons in the software. During the post-piracy modifications, the pirate may add additional fields at the end of the table structure or in between two existing fields in the table structure. Even if several modern data base management systems (DBMSs) provide techniques to introduce the new field logically in between two fields, say, between 4^{th} & 5^{th} fields, programmers usually tend to add the new field at the end of the table. Some DBMSs provide necessary software facility to add a new field without letting the user be bothered about the position of the new field in the database table and these tools usually place the new field at the end of the respective database table.

Often programmers either opt to add the additional field at the end of the table or simply don't care about inserting the additional field in the proper logical position. Some of them just use the software facility to add a new field and simply don't bother about where the software facility places the new field in the table. What is more important for the programmers is not the positioning of insertion of the additional field but the establishing of proper relationship. Irrespective of where the additional field is added (physically positioned), programmers can easily establish proper relationships or use proper SQL statements to display (or use) the additional field in any report generated by the software, logically, and in proper places. All these mean that even though post-piracy fields can be seen anywhere in the table, there are greater chances of finding them at the end of the table[5]. This also means that any difference found among the ending fields of the respective tables of the original and the pirated (or any successfully-unmapped fields at the end of the pirated table) can possibly be due to post-piracy modifications and so, special analysis of ending fields (with extra effort to identify and filter out post-piracy modifications) can yield reliable forensic result. For instance, Table-2 contains two unmapped fields UsrCode and UsrEnteredOn. . These two successfully-unmapped fields (see explanation on table-2 above) appear at the end of the database table in the pirated and so, the presence of these two fields (in the pirated) are to be further analysed.

External evidence of post-piracy modifications can exist: In some cases, in order to prove that a particular difference found was caused by post-piracy modifications and that the pre-modified version of the pirated had greater similarity with the original, the expert may require external supporting evidence, such as official documents. Log books (or documents for the software modifications done) and government directives (or documents initiating modifications in the software) belonging to post-piracy period are pieces of potential evidence acceptable to the court and the dates appear in these documents can be taken as evidence for post-piracy modification. In addition to log books and government directives, any official document that carries the date on which a particular new facility (say, a new MIS report) has been put to use by the client of the pirate, may be of help to the cyber forensic expert to argue unequivocally that the difference in the pirated is attributable to a post-piracy modification.

Summary of the discussion: In short, just as similarities need not always indicate piracy (Authors, 2009, p.176), differences need not always indicate non-piracy either. If, by establishing a schematically tangible patterning or mapping, the expert can identify the differences as attributable to post-piracy development, particularly suggestive of having been 'contrived', then the whole software comparison process may require closer attention along several parameters in different ways before rejecting or confirming piracy.

Need for judiciary-friendly presentation of results: The results of the analysis, when presented in transparent tabular form (to the judge) might provide more convincingly effective forensic evidence. It is hoped that Table-3 below, which is derived from Authors (2009, p.180), but specifically fine-tuned for post-piracy modifications, provides a seminal illustration for such a tabular presentation of, for instance, a database piracy forensics result[6]. Finally, it is needless to say that the result of analysis should be presented by cyber forensic expert as his/her views, strictly in an un-interpretive manner, because the right to interpretation solely rests with the court (Slade, 2004).

Further scope of this research: Needless to say, questionable differences between the pirated and the original can be found not only in data bases but also in other parts of the pirated, and for each part, along a variety of parameters. Some of the identifiable parts of software are source codes, databases, embedded images, fingerprints and so on. Again, one part can encode differences along several parameters. In case of source code, for instance, questionable differences can occur along parameters

[5] A statistical study to enumerate this chance is beyond the scope of this article.

[6] The test used in this example is "Substantial similarity" test (Davis, 1992). For other tests / approaches (for example, AFC), similar judiciary-friendly reports need to be arrived at.

like program variables, loop variables, names of functions, procedure calls, algorithms and so on. In the case of database, some of the parameters are field name, field type, field length and so on. In any case, post-piracy modifications along these parameters can make the software piracy investigation, delicate and demanding. This offers further scope in this research.

Table -3: The proposed format for presenting the result of comparison of two database tables (post-piracy modifications are also considered)

i.	**Similarity in the table names**: ___% commonality.
ii.	**Length of the 'original' table**: a
iii.	**Length of the 'pirated' table**: b
iv.	**Percentage of similarity in lengths**: (a/b)*100
v.	**Field count of the 'original'**: c
vi.	**Field count of the 'pirated'**: d
vii.	**Percentage of similarity in field count**: (c/d)*100
viii.	**Perfect commonality in the names of fields**: __ out of __ names of fields in the 'original' are found in 'pirated' also. So, ___% commonality
ix.	**Perfect commonality in name and data type among fields**: __ out of ___ fields have the common name and data types. So, ___% commonality in name and data type.
x.	**Perfect commonality in name, data type and length among fields**: ___out of ___ fields have same names, data types and length. So, ___% commonality
xi.	**Perfect commonality in name, data type, length and the default values set in the fields**: ___out of ___ fields have same names, data types, length and default values. So, ___% commonality.
xii.	**Perfect commonality in sequence of the fields with same name**: ___out of ___ fields with same name, do occur in the same sequence. So, ___% commonality.
xiii.	**Perfect commonality in sequence of the fields with same name, data type, length and default values**: ___out of ___ fields (with same name, data type, length and default values) do occur in the same sequence. So, ___% commonality.
xiv.	**Count of comparable (mappable) fields including suspected-post-piracy modified / created fields**: __ out of __ fields in the 'pirated' can be perfectly or approximately mapped (in terms of names) to at least one field in the 'original'. So, _____ % comparable fields in the 'pirated'.
xv.	**Count of non-mappable but suspected-post-piracy fields, including ending fields**: __ out of __ fields in the 'pirated' could not be properly mapped to any of the fields in the 'original' but they can be suspected to be post-piracy modifications. So, _____ % incomparable but suspected fields in the 'pirated'
xvi.	**Count of non-mappable, non-suspected fields**: __ out of __ fields in the 'pirated' could not be properly mapped to any of the fields in the 'original' and do not provide any clue to be suspected as post piracy modification. So, _____ % incomparable, non-suspected fields in the 'pirated'
xvii.	**Inference:** Piracy is confirmed / largely suspected / loosely suspected / not suspected.

7. CONCLUSION

To sum up, one can conclude that 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. In fact it is the differences that trigger the need for careful observation since they may be the result of post-piracy modifications. Such modifications may be motivated by a number of factors. A proper study of post-piracy modification, using the most appropriate tools both automatic and manual especially in the data base schemas of the pirated will unearth the differences that are invariantly attributable to post-piracy modifications, and thus will contribute substantially to the reliability of cyber forensic investigation. Some of elements discussed in this paper, like the positioning of the post-piracy fields and dates of post-piracy modification, are often incorrectly discounted as not too reliable; but under clever and careful handling, they can provide valuable supporting evidence. Ideally, in the interests of justice, a technical expert should be able to identify and put to use some or all the techniques of identifying post-piracy modifications to supplement the digital evidence (put forward by the automated tools, established judiciary approaches etc.) and other physical evidence.

8. REFERENCES

Authors, (2009), Software Piracy Forensics: Exploiting Nonautomated and Judiciary-Friendly Technique, Journal of Digital Forensic Practice, 2:4, 177 — 179

Author., (2002) Expert Commissioner Report submitted to the honourable court of Judicial I class magistrate, Kozhikode, Kerala, India, case number CMP 10371 / 2002, Software Associates vs. Together Infotech

Author, (2007), Seizure report submitted to the honourable District Court, Kozhikode, Kerala, India, on case number OS 2/2007, Sesame Software Solutions Vs. Perfect Software Solutions, p.2

Author, (2009), Judiciary-friendly computer forensics, Kerala Law Times, India, Part 13 & Index, 29th June, 2009, p.54

CDAC, (2008) Software Analysis Report number CDAC/RCCF/2007-20AR/Jan/2008 of the Resource Centre for Cyber Forensics, Centre for Development of Advanced Computing (CDAC), Government of India, Thiruvananthapuram – 695 033, Kerala, India, on the suit number OS 2/2007, Sesame Software Solutions Vs. Perfect Software Solutions, in the honourable District Court, Kozhikode, Kerala, India

Chang-Tsun Li, (2010) Handbook of Research on Computational Forensics, Digital Crime, and Investigation: Methods and Solutions, chapter XX, Information Science Reference, www.info-sci-ref.com,

ESALab (2007), The "SIMILE Workshop": Automating the detection of counterfeit software, available at www.esalab.com,

Frantzeskou, G., Stamatatos, E., Gritzalis, S., Chaski, C. E., and Howald, B. S., (2007) Identifying Authorship by Byte-Level N-Grams: The Source Code Author Profile (SCAP) Method, International Journal of Digital Evidence, 6, 1,

Hollaar, L. A., (2002), Legal Protection of Digital Information, BNA Books

Lancaster, T., and Culwin, F., (2004) A Comparison of Source Code Plagiarism Detection Engines, Computer Science Education, from http://www.informaworld.com/

Myles, G., (2006), Software Theft Detection through Program Identification, Ph. D. thesis, University of Arizona, Department of Computer Science, available at

http://sandmark.cs.arizona.edu/ginger_pubs_talks/defense_3_06.pdf

Reyes, A., (2007), Cyber Crime Investigations: Bridging The Gaps Between Security Professionals, Law Enforcement, and Prosecutors, Massachusetts, Syngress Publishing, Inc.

Slade, R. M., (2004), Software Forensics: Collecting Evidence From The Scene Of A Digital Crime, New York, The McGraw-Hill Companies, Inc.

van der Ejik P. (1994), Comparative Discourse Analysis of Parallel texts, eprint arXiv:cmp-lg/9407022, Digital Equipment Corporation, Ratelaar 38, 3434 EW, Nieuwegein, The Netherlands, CMP-lg/ 9407022,

Walker, J., (1996), Protectable 'Nuggests': Drawing the Line Between Idea and Expression in computer Program Copyright Protection, 44, Journal of the Copyright Society of USA, Vol 44, Issue 79

Yamamoto, T., Matsushita, M., Kamiya, T., and Inoue, K., (2004) Measuring Similarity of Large Software Systems Based on Source Code Correspondence, IEEE Transactions on Software Engineering, XX, Y (Proposed Draft found on http://www.google.com/url?sa=t&source=web&cd=3&ved=0CCwQFjAC&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.19.7035%26rep%3Drep1%26type%3Dpdf&rct=j&q=Measuring%20Similarity%20of%20Large%20Software%20Systems%20Based%20on%20Source%20Code%20Correspondence&ei=1SbTTZ-fEoLL0AGBjMn7Cw&usg=AFQjCNFc1Rip5lw3igaRe0o1yzI5pqDIlA&sig2=TMFVFzP6RfLRTbs4E42jSg&cad=rja