

نقدی بر استثناء مهندسی معکوس نرم افزارها در تحولات تقنینی ایران با مطالعه تطبیقی حقوق اتحادیه اروپا

تاریخ دریافت: ۱۳۹۸/۳/۱۳

تاریخ تأیید: ۱۳۹۸/۷/۸

سیدحسین شبیری (زنجان)^{*}

چکیده

دی کامپایل نرم افزار به دلیل این که مستلزم تکثیر، اقتباس یا ترجمه اثر اصلی است، طبق قواعد کلی نظام کپی رایت نقض حقوق پدیدآورنده اثر اصلی است. اما نظر به نیاز دیگر برنامه نویسان به خلق برنامه های سازگار و لزوم دی کامپایل نمودن آن به منظور کشف دانش برنامه جهت ایجاد برنامه مکمل یا سازگار، رهنمود ۱۹۹۱ اتحادیه اروپا در باره برنامه های رایانه ای و به تبع آن قوانین داخلی کشورهای اروپایی تحت شرایط خاصی دی کامپایل را مجاز اعلام نموده اند. لایحه قانون کپی رایت ایران (۱۳۹۳) نیز به پیروی از این رهنمود، دی کامپایل را جزء استثنائات حقوق مادی پدیدآورنده نرم افزار در نظر گرفته است. با وجود این، لایحه با الهام گیری ناقص از رهنمود فوق بسیاری از شروط مندرج در رهنمود را نادیده گرفته است. در این نوشتار با روش تحلیلی-توصیفی، بعد از تحلیل مفهومی دی کامپایل، شرایط مجاز بودن آن در مقررات اتحادیه اروپا و در حقوق ایران با یکدیگر مقایسه و به نقد کشیده شده است و ضمن تشریح مواد مربوط به دی کامپایل در لایحه قانون جامع مالکیت ادبی و حقوق مرتبط، خلاصه و ایرادات وارد بر آنها تبیین گردیده و سرانجام پیشنهادهای اصلاحی به قانونگذار ایران ارائه شده است.

واژگان کلیدی: مهندسی معکوس نرم افزار، استثنای حقوق تکثیر، نرم افزار سازگار و مکمل، اشکال زدایی نرم افزار، استفاده منصفانه، تفکیک ایده از بیان، اثر اقتباسی

* دانشیار گروه حقوق مالکیت فکری دانشکده حقوق دانشگاه قم (shshobeiri@yahoo.com).

مقدمه

روش «مهندسی معکوس» (Reverse Engineering) در مقابل مهندسی رو به جلو (Forward Engineering) فرایندی است که شخص با موضوع پژوهش قرار دادن کالا و محصول تولید شده صنعتی، در صدد یافتن فوت و فن محصول و دانش نهفته در آن و آگاهی از چگونگی ساخت و کارکرد و تعامل اجزای محصول صنعتی با یکدیگر می‌باشد (Mahajan, 1999, p.67). این روش در صنایع گوناگون به ویژه در صنعت نرم‌افزار کاربردهای فراوانی (از جمله به منظور ساخت برنامه سازگار و رفع اشکال‌ها و تصحیح خطاهای برنامه و ارتقاء برنامه) دارد. اصطلاح ویژه مفهوم یادشده در صنعت نرم‌افزار اصطلاح «دی کامپایل» (Decompile) است که در معنای دقیقش، خاص‌تر از مفهوم مهندسی معکوس می‌باشد (Laird, 1994, p.843). با وجود اهمیت این روش در صنعت مزبور، انجام این عمل چالش‌هایی را در نظام کپی‌رایت به همراه آورده است؛ زیرا به هنگام دی کامپایل نرم‌افزار رویدادهایی رخ می‌دهد که به حسب روال عادی، نقض کپی‌رایت دارنده نرم‌افزار است؛ اعمالی نظیر تکثیر، ترجمه و اقتباس از اثر همگی فرایندهایی هستند که در ضمن عمل دی کامپایل صورت می‌گیرد که طبیعتاً نیازمند کسب اجازه از پدیدآورنده و دارنده نرم‌افزار است، اما نظر به اهمیت این امر برای سازندگان نرم‌افزارهای رقیب و لزوم ایجاد تسهیلات قانونی و امکانات حقوقی برای این کار به دلیل توسعه و گسترش صنعت نرم‌افزار از طریق دی کامپایل نرم‌افزار و امکان افزایش خلق برنامه‌های سازگار و مکمل و اشکال زدایی از برنامه‌ها و نیز ارتقاء برنامه‌های موجود که در نهایت نفع حداکثری را برای عموم جامعه و مصرف‌کنندگان به دنبال خواهد داشت، قانونگذاران کشورهای پیشرو از جمله اتحادیه اروپا برآن داشته است که مقرراتی را برای جواز دی کامپایل به‌عنوان یکی از استثنائات حقوق مادی پدیدآورنده نرم‌افزار وضع نمایند. قانونگذار ایرانی در سابقه تقنینی پیشین خود (قانون ۱۳۷۹) هرچند به طور ناقص و در لایحه پیش رو باعنوان «قانون جامع مالکیت ادبی و هنری و حقوق مرتبط ۱۳۹۳» که هم اکنون مراحل تصویب نهایی اش را در مجلس می‌گذراند، به تبعیت از اتحادیه اروپا به صورت جامع‌تری بخش ویژه‌ای را به این امر اختصاص داده است. گرچه قانونگذار برای به روز نمودن قانون در این خصوص،

تلاش های فراوانی را صورت داده است، اما با بررسی دقیق مفاد این لایحه، با نقایص و خلاهای زیادی مواجه می شویم که ناشی از الگوبرداری ناقص از مقررات اتحادیه اروپا است. سؤال اصلی پژوهش جاری این است چگونه لایحه ایرانی توانسته است مقررات خود را نسبت به موضوع دی کامپایل نرم افزار مطابق با انتظارات این صنعت به روز نماید و آیا در این زمینه موفق بوده است؟ از این رو هدف پژوهش جاری این است که ابتدا با توصیف مقررات اتحادیه اروپا و نیز مقررات لایحه ایرانی و سپس با تجزیه و تحلیل این مقررات و مقایسه تطبیقی آنها، امتیازات و کاستی های لایحه و نیز در ضمن آن ایرادات حقوقدانان اروپایی بر مقررات اروپا تبیین گردد و از این رهگذر پیشنهادهایی به قانونگذار ایرانی جهت رفع نقایص موجود در لایحه داده شود، اما قبل از آن برای موضوع شناسی این مفهوم جهت خوانندگان لازم است مطالبی مقدماتی ارائه شود؛ براین اساس مباحث بخش اول این پژوهش به موضوع شناسی مهندسی معکوس اختصاص دارد. بخش اصلی و مهم این پژوهش مرتبط با تجزیه و تحلیل مقررات اروپایی در موضوع دی کامپایل نرم افزار و مقایسه مقررات لایحه ایرانی با این مقررات جهت بیان امتیازات و کاستی های لایحه از یک سو و ارائه پیشنهاد های لازم به قانونگذار ایرانی از سوی دیگر است که امید است قبل از تصویب نهایی لایحه موارد پیشنهاد شده مورد توجه و امعان نظر قانونگذار ایرانی قرار گیرد.

۱. مفهوم مهندسی معکوس نرم افزار

روش مهندسی معکوس که به معنای تجزیه و تحلیل کالای تولید و عرضه شده به منظور درک دانش و اسرار فنی نهفته در آن و آگاهی یافتن از چگونگی ساخت و تعامل اجزای کالا و محصول با یکدیگر است (Prestin, 2002, p.137)، در صنایع مختلف به ویژه صنعت نرم افزار کاربردهای فراوانی دارد (Mahajan, 1999, p.67). برای درک مفهوم این روش در نرم افزارها - که غالباً با تعبیر جایگزین «دی کامپایل» در برخی قوانین مالکیت فکری از آن یاد می شود (ر.ک: ماده 69e قانون کپی رایت آلمان ۱۹۹۸م) -، لازم است نگاهی کامل به کلیه مراحل ساخت یک برنامه کامپیوتری و متعاقب آن مهندسی معکوس آن داشته باشیم. فرض کنید برنامه رایانه ای الف توسط شرکتی ساخته

می‌شود. این برنامه غالباً با استفاده از یکی از زبان‌های رایج برنامه نویسی در قالب کدها و دستورات قابل فهم برای انسان یا همان «کد مبدا» (Source Code) نوشته می‌شود. سپس برنامه مزبور با استفاده از برنامه‌های تبدیل‌کننده برنامه به زبان ماشین - موسوم به کامپایلر (Compiler) - و در قالب کدهای غیر قابل فهم برای انسان به صورت صفر و یک یا «کد مقصد» (Object Code) تبدیل می‌شود و به صورت برنامه اجرایی ب به مشتریان عرضه می‌شود. هنگام عرضه به خریدار غالباً موافقتنامه لیسانس استفاده بین دارنده کپی‌رایت و خریدار امضا می‌شود. حال خریدار یا شخص ثالث ممکن است با بهره‌گیری از روش مهندسی معکوس، برنامه اجرایی و گذشته ب را به صورت معکوس و وارونه دی کامپایل کرده و از آن برنامه اصلی جیم را استخراج نماید و سپس با تجزیه و تحلیل برنامه جیم و یافتن رازهای نهفته و فناوری‌های برنامه اقدام به نوشتن برنامه جدید دال نماید (Jeong, 2006, pp.140-141).

باتوجه به مطالب یادشده ضرورت مهندسی معکوسی معکوس در نرم افزار در جایی که سازنده نرم افزار مستندات فنی و کد اصلی برنامه را منتشر نساخته است مشخص می‌شود؛ زیرا برنامه ب که در اختیار کاربران قرار داده می‌شود در قالب کدهای مقصد است که تنها برای رایانه قابل فهم است و نمی‌توان از آن طریق، اطلاعات و دانش فنی نهفته شده در اثر را به دست آورد مگر اینکه به صورت معکوس برنامه دی کامپایل گشته و به زبان قابل فهم برای انسان تبدیل شود تا اسرار موجود در برنامه کشف و استخراج شوند (Goldstein, 2007, p.7).

شایان گفتن است درباره مفهوم یادشده در مستندات حقوقی به اختصار می‌توان گفت در رویه قضایی امریکا، دادگاه منطقه ایالات متحده به هنگام بررسی پرونده‌ای با موضوع مهندسی معکوس در سال ۱۹۸۹، مهندسی معکوس را این چنین تعریف کرده است: «فرآیند شروع با محصول کامل شده و کارکردن معکوس با آن برای تجزیه و تحلیل آن با این هدف که محصول چگونه کار میکند و چگونه ساخته شده است» (Behrens & Levary, 1997, p.31).

۲. چگونگی نقض کپی‌رایت با مهندسی معکوس و چالش‌های

حقوقی ناشی از آن

حال باتوجه به فرایندهای یادشده در فرآیند مهندسی معکوس، مسئله حقوقی پیش روی این است که کدام بخش از اعمال فوق نقض کپی‌رایت به حساب می‌آید؟ نخست اینکه آیا برنامه تولید شده دال نقض کپی‌رایت برنامه الف یا ب است و دوم اینکه آیا برنامه جیم کپی‌رایت برنامه الف را زیرپا گذاشته است؟ در وهله اول در پاسخ به سؤال اول که آیا دال می‌تواند ناقض کپی‌رایت الف یا ب باشد، باید گفت این امر وابسته به قواعد کلی نظام کپی‌رایت است؛ یعنی دال را به شرطی ناقض الف می‌توان دانست که اولاً شباهت اساسی با الف داشته باشد (نقض حق تکثیر) و ثانیاً دال بدون هیچ خلایقی صرفاً از الف برگرفته شده باشد (فقدان شرط اصالت) (Sugiyama, 1999, p.12).

در وهله دوم باید به این سؤال پاسخ داده شود که تجزیه و تحلیل و انتزاع برنامه جیم از برنامه ب آیا نقض حق تکثیر برنامه الف یا ب تلقی می‌شود یا خیر؟ در پاسخ به این مسئله باید گفت گرچه برنامه جیم با دی کامپایل از برنامه ب اقتباس شده است، اما جملات و دستورات برنامه جیم کاملاً متفاوت با برنامه الف می‌باشد. به همین جهت برخی بر این باورند که به علت عدم شباهت لفظی بین عبارات برنامه جیم و برنامه الف علی‌رغم شباهت معنایی و عملکردی نباید آن را در مقوله تکثیر اثر پیشین جای داد. بلکه آن را باید مصداق خلق اثر جدید اقتباسی دانست. با وجود این از منظر دیگری می‌توان عنوان تکثیر اثر را برای محصول جیم در نظر گرفت؛ زیرا هیچ فعالیت ذهنی جدید و خلاقانه‌ای جهت خلق اثر جیم انجام نپذیرفته است و نمی‌توان آن را به مانند دیگر آثار ترجمه‌ای که مترجم خلاقیت‌هایی را از خود بروز می‌دهد دانست. پس به خاطر شباهت اصلی و همانندی غیر لفظی بین الف و جیم آن دو را باید مصداق تکثیر اثر قلمداد کرد. به همین خاطر، برخی از دادگاه‌ها در ژاپن در دعوی میکروسافت در سال ۱۹۸۷ این عمل را نقض حق تکثیر به حساب آورده‌اند (Tokyo District Court, 1987, no.1219).

۳. سیر تحول مهندسی معکوس نرم افزار در کشورهای غربی

در دهه ۱۹۵۰ که صنعت نرم افزار در ابتدای راه خود بود و دوران طفولیت خود را طی می کرد، در مهد صنعت نرم افزار - یعنی دره سیلیکون در ایالات متحده امریکا - خطاب برنامه به جای سازندگان برنامه ها عموم مردم بود. براین اساس، دیدگاه به اشتراک گذاری متن برنامه ها عرف و هنجاری عمومی بود و برنامه نویسان به صورت باز و بدون محدودیتی ایده ها و افکار خود را با یکدیگر در میان می گذاشتند (Carver, 2005, p.443). به موازات گسترش استقبال از فناوری های رایانه ای نه تنها در امریکا بلکه در اروپا و دیگر کشورها نیز سازندگان نرم افزار متوجه منافع گسترده اقتصادی نرم افزار شدند و از اینجا بود که به تدریج شروع به کنترل گسترده بر روی کدهای برنامه شان از راه های مختلف نمودند. صنعت نرم افزار امریکا مسیر خود را به سوی ارائه مدل های بسته نرم افزاری با تبدیل برنامه به کدهای غیر قابل فهم و رمزآلود سوق داد و به موازات آن نظام حقوقی امریکا به عنوان کشور دارای نظام کامن لا تلاش نمود تا مسئله دی کامپایل یا مهندسی معکوس نرم افزار را از طریق مکانیسم «نظریه استفاده منصفانه» (Fair Use) مندرج در ماده ۱۰۷ قانون کپی رایت امریکا حل و فصل نماید (Blythe, 2006, p.111). بدیهی است واگذاری مکانیسم مزبور و احراز وجود عناصر چهارگانه استفاده منصفانه مندرج در ماده مزبور از طریق قاضی دادگاه، آزادی و اختیار عمل زیادی را به دادگاه ها می داد و نتیجه آن ایجاد طیفی بود که در یک سوی آن، تفسیر موسع نظریه فوق قرار داشت که استفاده کنندگان نرم افزار را بدون هیچ محدودیتی و بدون هیچ محدودیتی رها می ساخت تا آنها بتوانند با کمترین هزینه و تلاش، زحمت رقیبان سازنده نرم افزار دی کامپایل شده را به یغما ببرند و در سوی دیگر طیف، تفسیر مضیق دکترین استفاده منصفانه بود که به سازندگان نرم افزارها قدرتی انحصاری نسبت به دانش و فوت و فن های نهفته در نرم افزارهایشان می داد و در نتیجه مانع توسعه علم و گسترش دانش نرم افزار از یک سو و توسعه نرم افزارهای جدید سازگار می شد (Donna, 2006, pp.537-555). اتحادیه اروپا هم که در شرف ایجاد هویتی جدید در صنعت نرم افزار در سطح جهانی بود، توانسته بود رقیبی جدی امریکا در بازار جهانی نرم افزار گردد. این تلاش ها سرانجام در رهنمود ۱۹۹۱ اتحادیه اروپا در موضوع برنامه های

رایانه‌ای تجلی پیدا کرد و با پیش‌بینی استثنای بسیار بحث‌انگیز دی‌کامپایل برنامه در ماده ۶ این رهنمود توانست مدل منحصر به فرد و جدیدی را به جهان ارائه دهد و با شگردی ظریف و دقیق، در مرز باریک بین منافع عموم استفاده‌کنندگان و منافع خصوصی سازندگان نرم‌افزار در مهندسی معکوس نرم‌افزارها حرکت نماید. استثنای دی‌کامپایل نرم‌افزار در این ماده می‌تواند به عنوان تابلو و نمادی برای بیان نظر اتحادیه اروپا نسبت به مسئله دی‌کامپایل در جهانی که به‌سوی سیستم‌های باز نرم‌افزاری حرکت می‌کند به حساب آید و همانطوری که می‌آید، این ماده موضوع «سازگاری نرم‌افزار» را به عنوان معیاری اساسی و شرط ضروری نسبت به مجاز شمردن دی‌کامپایل نرم‌افزار در اتحادیه اروپا در نظر گرفته است (Heindl, 2008, pp.204-207).

۴. وضعیت مهندسی معکوس در اتحادیه اروپا

همانطوریکه گذشت، در پاسخ به مسئله دی‌کامپایل نرم‌افزار رهنمود اتحادیه اروپا در ماده ۶ خود چهارچوب مشخصی را تعریف و تبیین نموده است. علیرغم بحث‌های گسترده پیرامون موضوع دی‌کامپایل بین قانونگذاران و سیاستگذاران حقوقی اروپا، سرانجام مصالحه‌ای بین بازیگران اصلی بازار در صنعت نرم‌افزار و برنامه‌نویسان و کاربران صورت گرفت و رژیم حقوقی پالایش شده‌ای در ماده ۶ رهنمود پیش‌بینی شد. این ماده در حقیقت استثنایی بر حقوق انحصاری پدیدآورنده نرم‌افزار است و به همین خاطر از ابعاد گوناگون باید تفسیر مضیق شود. هر برنامه کامپیوتری به مثابه یک جعبه سیاه در نظر گرفته می‌شود که نگاه کردن به داخل آن مستلزم کپی اجزای آن است و بدیهی است که در نظام کپی‌رایت نیازمند کسب اجازه از دارنده است، اما طبق رهنمود مزبور در دو مورد لزوم تحصیل اجازه تخصیص و استثنا زده شده است و می‌توان آن را بدون نیاز به کسب اجازه دی‌کامپایل کرد؛ نخست برای تصحیح اشتباهات (ماده ۵) و دوم برای ایجاد سازگاری (ماده ۶). اصولاً دستیابی به فناوری‌های به کار رفته در برنامه در نظام کپی‌رایت بر خلاف نظام اسرار تجاری و اختراعات اشکالی ندارد و آنچه که در این نظام تخلف به حساب می‌آید تکثیر اثر به عنوان نقض حق انحصاری پدیدآورنده است (Rotenberg, 2005, p.15).

رهنمود اروپا برخلاف امریکا رویکرد استفاده منصفانه را در پیش نگرفته است، اما در عوض به صورت مستقیم، مسئله دی کامپایل را در قانون مورد توجه صریح خود قرار داده است (Jaburek, 1993, p.21). طبق بند یک ماده ۶ عمل تکثیر اثر یا ترجمه و تبدیل آن به زبان قابل فهم برای انسان نیازمند کسب اجازه از دارنده برنامه دی کامپایل شده نیست به شرطی که این امر برای کسب اطلاعات لازم جهت خلق اثر نرم‌افزاری مستقل و سازگار با برنامه‌های دیگر ضروری و لاینفک باشد. ایده و اندیشه ورای ماده ۶ این است که دی کامپایل برنامه اصولاً مجاز نمی‌باشد مگر در موارد معین و کاملاً محدود و مشخص. علاوه بر ماده ۶، ماده ۲۲ رهنمود هم این امر را تأکید می‌کند که دی کامپایل فقط تحت شرایط بسیار محدودی مجاز می‌باشد و بنابراین بی جهت نیست که در ترجمه هلندی رهنمود، عبارت «شرایط محدود» به «شرایط نادر» تعبیر شده است (Jongen & Meiboom, 1993, p.15).

ماده ۶ موضوع دی کامپایل را در سه فراز مورد عنایت خود قرار داده است: «اهداف، شرایط و استفاده‌های مجاز» اما قبل از پرداختن به جزئیات، لازم است متن ماده آورده شود:

ماده ۶ دستورالعمل اروپایی حمایت از برنامه‌های رایانه ای: «۱. در مواردی که تکثیر برنامه مبدأ (Source Code) یا ترجمه ظاهر برنامه به مفهوم قسمت‌های الف و ب بند ۱ ماده ۴ برای دسترسی به اطلاعات لازم برای هم کنشی یک برنامه ایجاد شده به طور مستقل با سایر برنامه‌ها ضرورت داشته باشد به شرط اجتماع شرایط زیر نیاز به اجازه صاحب حق ندارد: الف) این اعمال توسط انتقال گیرنده حق بهره برداری یا توسط هر شخص دیگری که حق استفاده از کپی برنامه را دارد یا توسط شخصی که به حساب آنها عمل می‌کند انجام گیرد؛ ب) اطلاعات لازم برای هم کنشی، برای اشخاص یادشده در بند الف به سادگی و به سرعت قابل دسترسی نباشد؛ ج) این اعمال محدود به بخش‌هایی از برنامه اصلی که برای هم کنشی ضروری است محدود شود.

۲. مقررات بند ۱ فوق، بهره برداری از اطلاعات به دست آمده در نتیجه اعمال این بند را در موارد زیر توجیه نمی‌کند: الف) برای اهدافی غیر از ایجاد هم کنشی برنامه رایانه‌ای خلق شده به طور مستقل مورد استفاده قرار گیرد. ب) جز در مواردی که برای

هم کنشی برنامه رایانه‌ای خلق شده به طور مستقل به اشخاص ثالث ارایه شود. ج) برای اجرا، تکثیر یا تجاری کردن برنامه رایانه‌ای که محتوای آن به طور اساسی مشابه است یا برای هرگونه عمل دیگر که به حقوق پدیدآورنده آسیب وارد سازد، مورد استفاده قرار گیرد»

۴-۱. اهداف مجاز

ماده ۶ دی کامپایل را فقط با هدف ایجاد «برنامه مکمل و سازگار» مجاز دانسته است. کمیسیون اروپا در بیستمین گزارش خود در موضوع سیاستگذاری نظام رقابت اعلام نموده است برنامه‌ای که از طریق دی کامپایل به وجود می‌آید می‌تواند با برنامه اصلی کامپایل شده نیز رقابت کند و در این چنین مواردی معمولاً ارتباطی با آن نخواهد داشت (Lai, 2000, p.100). یکی از اعضای کمیسیون آقای ساکر (Sucker) تأکید می‌کند که چه بسا شخص می‌تواند با مهندسی معکوس، برنامه سازگاری تولید کند که در بازار با برنامه اصلی نیز رقابت کند و حتی جایگزین آن شود (Sucker, 1993, p.17).

اما مستفاد از ماده ۶ این است که رهنمود اجازه دی کامپایل نرم افزار را فقط برای هدف سازگاری می‌دهد؛ بنابراین مفهوم این مقرر این خواهد بود که شخص نمی‌تواند برای اهداف غیر مرتبط با سازگاری نرم افزار نظیر پژوهش و تحقیق نسبت به نحوه عملکرد کدها و ایده‌های نهفته در آن، برنامه را دی کامپایل کند؛ بنابراین محور اصلی شرط مجاز بودن مسئله «سازگاری» است که لازم است در ادامه، ابعاد مختلف آن موشکافی و تشریح شوند.

۴-۲. مفهوم و اهمیت سازگاری در صنعت نرم افزار

در پاره‌ای از نوشته‌ها مفهوم سازگاری اینگونه تعریف شده است: «اتصال درونی و به هم پیوستگی داخلی منطقی و در موارد مقتضی فیزیکی به این منظور که اجازه دهد همه عناصر یک نرم افزار و یا سخت افزار با دیگر نرم افزارها و سخت افزارها کار کند» (Heindl, 2008, p.33) برای درک ماهیت و اهمیت سازگاری ممکن است تفکیک دو سطح سازگاری نرم افزار به شکل زیر مفید باشد:

۱. نرم افزار سیستم عامل؛ دربردارنده برنامه‌هایی است که عملکردهای سطح پایین لازم برای کار با سخت افزار کامپیوتر را فراهم می‌کند. نرم افزار سیستم عامل، برنامه‌های پشتیبان و ابزار اصلی خلق برنامه مثل برنامه کامپایلر، برنامه تست و اشکال زدایی برنامه، برنامه سرویس‌های عمومی مثل مدیریت فایل، انتقال فایل و فرمت حافظه جانبی دستگاه.

۲. نرم افزارهای کاربردی؛ برنامه‌هایی هستند که با هدف انجام کارهای خاصی، طراحی و پیاده سازی شده اند، مثل پردازشگر کلمه، صفحه گستر، بانک‌های اطلاعاتی. به عنوان نمونه، برنامه سیستم عامل ویندوز شرکت مایکروسافت یا برنامه سگا بسترهایی هستند که بر روی آن نرم افزارهای کاربردی طراحی می‌شوند تا در آن محیطها اجرا شوند (برای مثال برنامه لوتوس، برنامه بازی سگا). برای اینکه نرم افزار کاربردی با سیستم عامل در ارتباط و تعامل باشد، برنامه نویس رایانه‌ای باید دسترسی به اطلاعات رابط نرم افزار سیستم عامل داشته باشد که به آن رابط‌های کاربری یا اینترفیس برنامه کاربردی ("Application Programming Interface "API") گفته می‌شود (Howard & Gushow, 1995, p.24). در واقع، اینترفیس برنامه اطلاعاتی هستند که عملیات سازگاری را انجام می‌دهند (Samuelson & Scotchmer, 2002, pp.165-166).

۳-۴. پیشینه سازگاری و راهبردهای مربوط در صنعت نرم افزار

در دهه ۸۰ شرکت سخت افزاری ای بی ام (IBM) اولین محصول کامپیوتر شخصی (PC) را عرضه نمود که قلب آن بخش بایوس (BIOS) آن بود که مربوط به سیستم ورودی و خروجی ای می‌شد که اجزای سخت افزاری رایانه - مثل درگاه (پورت) چاپگر، صفحه مانیتور و صفحه کلید - را به کار گرفته و به راه می‌انداخت؛ بنابراین هر برنامه نویسی برای سازگار شدن برنامه اش با سخت افزار رایانه‌ای بی ام لازم بود بایوس آن را کپی کند. با وجود این شرکت ای بی ام با شرکت نرم افزاری میکروسافت همکاری مهمی را آغاز نمود که منجر به تولید سیستم عامل بسیار مشهور داس (DOS) شد که به عنوان بستری نرم افزاری برای اعمال و اجرای برنامه‌های رایانه‌ای بر روی رایانه‌های شخصی ای بی ام بود. به عبارت دیگر، برای تحت کنترل درآوردن سخت

افزار کامپیوتر شخصی، نرم افزار کاربردی مجبور بود با سیستم عامل داس از طریق رابط‌های کاربری موسوم به ای. پی. ای (A.P.I) در تعامل باشد. از سوی دیگر تهیه‌کنندگان نرم افزار در ارتباط با مسئله سازگار شدن برنامه‌های دیگران با برنامه‌هایشان راهبردهای متفاوتی را در پیش می‌گیرند؛ برخی از آنها رابط‌ها یا اینترنت‌های را به همراه برنامه منتشر ساخته و آن را به فروش می‌رسانند؛ برخی اجازه استفاده از لیسانس‌ها را به صورت رایگان یا به ازای پرداخت پول به دیگران می‌دهند و در نهایت برخی رابطه کاربری را به صورت اسرار تجاری مخفی نگاه داشته و آن را ارائه نمی‌دهند. بدیهی است هدف برخی از مخفی نگه داشتن رابط‌ها، تسلط و کنترل و تسخیر بازار است، اما همان طوری که در دعوی دولت امریکا علیه شرکت میکروسافت (United States v. Microsoft, 65 F. Supp. 2d 1, 15 (D.D.C. 1999)) مطرح شد، کنترل و پنهان سازی و انحصار در افشاء رابط‌های کاربری برنامه به معنای کنترل و انحصار صنعت نرم افزار است که خلاف اصول رقابت به حساب می‌آید (Vinje, 1992, pp.250-259).

۴-۴. انواع سازگاری در صنعت نرم افزار

به طور معمول، آن دسته از تهیه‌کنندگانی که برنامه را در اختیار دیگران قرار می‌دهند، دو هدف متفاوت از این امر را دنبال می‌کنند: ۱. سازگاری افقی یا عرضی؛ ۲. سازگاری عمودی یا طولی که در ادامه به آنها می‌پردازیم.

۴-۴-۱. مفهوم سازگاری افقی یا هم عرض

سازگاری افقی به معنای در دسترس قرار دادن اطلاعات رابط یک سیستم عامل (اینترفیس آن) به گونه‌ای که این امکان را برای مهندس معکوس رقیب فراهم سازد تا سیستم عامل جدید خود را خلق کند در حالی که این سیستم عامل با دیگر برنامه‌های کاربردی موجود در سیستم عامل اولی سازگار باشند. به دیگر سخن، سازگاری افقی منجر به توسعه سیستم عامل رقیبی می‌شود که سازگار با برنامه‌های کاربردی سیستم عامل اولی خواهد بود. بدیهی است این سیستم عامل جدید خود با سیستم عامل قبلی

سازگار نیست؛ زیرا قرار نیست که در آن سیستم عامل قبلی اجرا شود و کار کند. اگر بخواهیم مثالی برای این مورد ارائه شود می‌توان فرضی را در نظر گرفت که شرکت سخت افزاری ای بی ام اطلاعات و رابط‌های کاربری بایوس سیستم عامل داس خود را برای اینکه بتواند برنامه‌های دیگر در محیط سیستم عامل‌های مختلف با آن سازگار شوند را ارائه می‌دهد تا به وسیله آن شرکت‌های نرم‌افزاری رقیب مثل اپل (Apple) بتوانند بایوس خود را به گونه‌ای به وجود آورند که همه برنامه‌های کاربردی که با بایوس ای بی ام نوشته شده‌اند مثل نرم‌افزار لوتوس (Lotus) و ورد پرفکت (Word Perfect) قادر باشند در این سیستم عامل رقیب اپل نیز کار کنند. با این ترفند برای برنامه نویسان کاربردی، انگیزه‌ای قوی ایجاد می‌شود که برنامه‌های خود را در محیط‌های سیستم عامل ای بی ام (یعنی سیستم عامل داس) بنویسند چرا که این نرم‌افزارها در محیط‌های سیستم عامل دیگر مثل اپل نیز کار خواهند کرد و در نتیجه تفوق برنامه نویسان بر بازار در محیط سیستم عامل‌های مختلف نیز حفظ خواهد شد (Heindl, 2008, p.35).

نمونه دیگری از سازگاری افقی در ایالات متحده امریکا در قضیه شرکت سونی (Sony) اتفاق افتاد. در آنجا دسترسی افقی از طریق مهندسی معکوس به رابط کاربری سونی سبب شد که نرم افزارهای سازگاری نوشته شوند که بتوانند هم در محیط سخت افزاری پلی استیشن (PlayStation) کار کنند و هم در محیط رایانه‌های معمولی (Sony Computer Entertainment, Inc. v. Connectix Corp., 2000).

۴-۴-۲. کاربرد و فایده سازگاری افقی

از منظر کارایی اقتصادی، تهیه یک سیستم عامل جدید برای یک کامپیوتر شخصی (Personal Computer) عملی مشکل، زمان‌بر و همراه با خطرهای فراوان سرمایه‌گذاری اقتصادی است. با وجود این چرا مهندسی نرم افزار تلاش خود را معطوف می‌سازند تا وارد این بازارهای افقی و موازی شوند؟ پاسخ روشن است؛ زیرا در غیر اینصورت با خلق سیستم عامل ناسازگار جدید هیچ برنامه کاربردی که قبلاً در محیط سیستم عامل قبلی کار می‌کرده است نمی‌تواند بر روی این سیستم عامل جدید اجرا

شود و این مانعی جدی برای سیستم عامل نویس‌های جدید خواهد بود که وارد این بازار شود؛ زیرا کاربرها بسیار بعید است سیستم عاملی را تهیه کنند که قادر نیست نرم افزارهای کاربردی که قبلاً بر روی سیستم عامل پیشین اجرا شده و مراحل تست خود را گذرانده و با آن آشنا شده‌اند را بر روی این سیستم عامل جدید نیز اجرا نماید (Rotenberg, 2005, pp.11-13).

با وجود این، سازگاری افقی این فایده را به همراه خواهد داشت که سیستم عامل‌های جدیدی خلق شوند که با نرم افزارهای کاربردی موجود سازگاری داشته باشند، به ویژه نرم افزارهایی که با محیط سیستم عامل داس سازگار هستند. پرواضح است این امر انگیزه‌ای قوی برای انجام فرآیند هزینه‌بر و پرخطر دی کامپایل برای مهندسين نرم افزار نسبت به سیستم عامل قبلی جهت کشف رابط کاربری آن فراهم خواهد نمود (Heindl, 2008, p.35).

۴-۴-۳. سازگاری عمودی یا در طول

سازگاری عمودی در جایی اتفاق می افتد که تهیه کننده سیستم عامل اطلاعات رابط کاربری خود را در اختیار سازندگان نرم افزارهای کاربردی قرار می دهد تا آنها بتوانند به صورت مستقل خود برنامه‌های کاربردی‌ای را خلق کنند که بر روی این سیستم عامل قابل اجرا باشد. اتخاذ این چنین سیاستی برای شرکت‌های کامپیوتری نظیر میکروسافت سبب شد که سیستم عامل داس این شرکت سیستم عامل استاندارد در آن برهه از زمان گردد و بدینسان سیستم عامل‌های سازگار دیگری نظیر دی. آر. داس. خلق شد که همان کارکرد رابط‌های داس را دارند و لذا برنامه‌های کاربردی موجود در سیستم عامل داس در آن محیط هم قابل اجرا می‌باشند. به عبارت دیگر به جهت اینکه سیستم عامل دومی قابل جایگزینی با اولی بود، این دو سیستم عامل رقیب هم گشتند (Rotenberg, 2005, p.13).

مسئله سازگاری عمودی در پرونده سگا در امریکا مطرح شده است. در قضیه سگا، سازنده رقیب شرکت سگا تصمیم می‌گیرد با استفاده از فرآیند مهندسی معکوس، اطلاعات حمایت شده در کنسول بازی سگا را کپی کند تا بدین وسیله بتواند بازی‌هایی

را تولید کند که بتوانند در محیط سگا قابل اجرا باشد. بطور کلی، سازگاری عمودی با یک محیط خاص این تأثیر اقتصادی مثبت را برای آن محیط به وجود می‌آورد که سبب افزایش تعداد برنامه‌های قابل اجرا در آن محیط گشته و پرواضح است این امر ارزش اقتصادی آن محیط کامپیوتری را بالا می‌برد (Sega Enterprises, Ltd. v. Accolade Corp., (1993).

۴-۴-۴. مقایسه سازگاری افقی با عمودی

در مقایسه سازگاری افقی با سازگاری عمودی، باید گفت حمایت قانونی از سازگاری افقی این خطر و تهدید را برای سیستم عامل نویس‌های مشهور در بازار که رابط کاربری شان را در اختیار دیگران قرار می‌گیرد به همراه خواهد داشت که بازار کامپیوتر در اختیار رقبایشان قرار خواهد گرفت و بستری برای سیستم عامل‌های رقیبی که قانوناً مهندسی معکوس را انجام داده‌اند فراهم خواهد ساخت. از این رو سیستم عامل نویسان مسلط بر بازار معمولاً تمایلی به سازگاری افقی ندارند و در عوض روش سازگاری عمودی که از این جهت بی‌خطر است را دنبال می‌نمایند (Petra Heindl, id, (p.37).

۴-۵. گستره و قلمرو سازگاری در رهنمود اروپا

به نظر می‌رسد فارغ از آنچه که در کمیته‌های فنی و حقوقی مربوطه مطرح شده، رهنمود برنامه‌های رایانه‌ای اروپا هم سازگاری افقی و هم عمودی را پذیرفته و هر دو را مشمول حمایت خود قرار داده است. با عنایت به اینکه عنوان ماده ۶ رهنمود دی کامپایل است، احتمالاً نمی‌توان این را به گونه‌ای تفسیر نمود که شامل تمامی مصادیق مهندسی معکوس شود. پیرو بند یک ماده ۶ رهنمود دی کامپایل باید امر ضروری و شرط لازم برای کسب اطلاعات لازم برای رابط برنامه باشد تا بتوان نرم افزار سازگار با برنامه دی کامپایل شده نوشته شود؛ بنابراین یکی از احتمالاً مهمترین هدف ماده ۶ این است که هدفی غیر از هدف سازگاری مانند تحقیق نباید مورد نظر دی کامپایل کننده باشد، اما اینکه آیا واقعا هدف قانونگذار از ذکر سازگاری محدود نمودن دی کامپایل با

هدف سازگاری است یا اینکه این امر اصولاً از غفلت او ناشی شده و نظری نسبت به مقاصدی همچون تحقیق نداشته، خیلی روشن و آشکار نیست، به ویژه اینکه در ایالات متحده مهندسی معکوس برای اهداف تحقیق چنانچه دارای شرایط استفاده منصفانه باشد روا و جایز شمرده شده است (ماده ۱۰۷). لذا برخی از نویسندگان معتقدند که در این ماده، دی کامپایل با اهدافی نظیر تحقیق اصلاً مورد لحاظ قانونگذار نبوده و به اصطلاح این ماده اطلاق از این جهت ندارد و منصرف از این مورد است (Czarnota & Hart, 1994, pp.329-337). همانگونه که یکی از مفسران رهنمود در دفاع از این نظریه اظهار کرده است، ممنوع کردن فعالیت‌های پژوهشی در حوزه توسعه نرم‌افزار این ماده را در معرض نقض ماده سه معاهده اتحادیه اروپا قرار می‌دهد که بیان داشته که هدف اتحادیه ارتقا و توسعه فناوری است^۱. به علاوه رهنمود نسبت به اینکه هدف سازگاری در ماده نه تنها شامل دی کامپایل برای خلق محصولات مکمل و سازگار می‌شود بلکه برای خلق محصولات جایگزین جهت ایجاد رقابت بیشتر در بازار می‌شود، ساکت است. موافقان این ماده در تأکید بر تصویب آن تمایل مصرف‌کنندگان و شرکت‌های کوچک نرم‌افزاری را نسبت به اهمیت استثناکردن دی کامپایل را خاطر نشان می‌ساختند تا آنها بتوانند قدرت خلق نرم‌افزارهای رقیب را داشته باشند.

رهنمود قلمرو سازگاری را به طور گسترده تعریف کرده است؛ یعنی امکان برقراری هرگونه ارتباط داخلی همه عناصر نرم‌افزاری مختلف جهت تعامل با یکدیگر، اما در جای دیگر سازگاری را به گونه دیگر تعریف کرده است: «توانایی تبادل اطلاعات با یکدیگر» بنابراین هدف کلی از دی کامپایل امکان ارتباط بین اجزای سیستم کامپیوتری شامل سیستم‌هایی که مربوط به سازندگان دیگر است می‌شود. به هر حال، وجود این تعاریف ناکافی از مفهوم سازگاری در رهنمود، استثنا مزبور را از لحاظ قلمرو مبهم ساخته است. مشابه همین ابهام عبارتی در نظام حقوق انگلستان نیز قبلاً اتفاق افتاده بود که پیشنهاد شده بود دی کامپایل مجاز فقط محدود به مواردی شود که هدف، تولید برنامه مرتبط و سازگار است و نه تولید برنامه رقیب؛ چرا که دی کامپایل بر اساس

۱. ماده ۳ معاهده تأسیس اتحادیه اروپا ۲۰۰۲ این گونه مقرر نموده است: «فعالیت‌های اتحادیه اروپا شامل موارد زیر می‌شود: ... مورد a) ارتقای تحقیق و توسعه فناوری».

مبنای استفاده منصفانه توجیه می‌شود و نه بر اساس اهداف اقتصادی؛ بنابراین توسعه نرم‌افزار رقیب مبتنی بر اطلاعات دی کامپایل شده نباید هدف مجاز دی کامپایل به حساب آید، اما علیرغم آنچه که در انگلستان گذشت متعاقب استدلال‌های موافق و مخالف در پارلمان اروپا کمیسیون اتحادیه اروپا با مشورت با پارلمان به این نتیجه رسید که اصطلاح سازگاری را باید به گونه‌ای تفسیر نمود که هم شامل نرم‌افزارهای سازگار شود و هم شامل نرم‌افزارهای رقیب؛ بنابراین چه بسا برنامه ایجاد شده از طریق دی کامپایل ممکن است با برنامه دی کامپایل شده که برای هدف سازگاری است در رقابت هم باشد بدون اینکه با اشکال و نقدی همراه باشد (Marly, 1995, p.294).

مسئله دیگری که در کمیته اروپایی مطرح شد این بود که آیا هدف سازگاری می‌تواند شامل ساخت برنامه‌های جدید سازگاری باشد که هدف آن فقط تبادل و به اشتراک گذاری اطلاعات با برنامه دی کامپایل شده باشد یا اینکه باید سازگاری به گونه‌ای صددرصد باشد یعنی برنامه جدید تمامی عملیات پیش بینی شده در برنامه دی کامپایل شده را دقیقاً به همان شیوه انجام دهد (Walter, 2001, p.221)، اما به نظر می‌رسد این معنای دوم را نمی‌توان از عبارت ماده ۶ استفاده کرد بنابراین برای تعیین اینکه چگونه و تا چه گستره‌ای مفهوم سازگاری را باید درک و تفسیر کرد همه این‌ها وابسته به مسائل موضوعی است که دادگاه‌ها در هر مورد به طور خاص نظر خواهند داد و این امر هدف قانونگذار اتحادیه اروپا که ایجاد وحدت بین کشورهای عضو است را با امکان تفاسیر مختلف از سوی دادگاه‌ها تضعیف خواهد کرد (Guillou, 1998, p.533).

بنا به نظر دو تن از حقوقدانان، ماده ۶ تنها اجازه دی کامپایل برنامه برای سازگاری بین دو نرم‌افزار با یکدیگر را می‌دهد و بنابراین شامل دی کامپایل با هدف سازگاری بین سخت افزار و نرم‌افزار نخواهد بود؛ زیرا رهنمود به صراحت سازگاری با سخت افزار را به عنوان اهداف مجاز دی کامپایل اعلام نکرده است (Czarnota & Hart, 1994, pp.84-86)، اما با وجود این ادعا، بدیهی است که اطلاق مفهوم سازگاری در این ماده شامل سازگاری با سخت افزار هم می‌شود و به نظر می‌آید هدف اصلی ماده اجازه رقابت در تمامی سطوح است و ادعای اشخاص یادشده مبنی بر کشیدن خطی میان

سازگاری نرم افزار با نرم افزار و سازگاری نرم افزار با سخت افزار امری ناخوشایند و ناصحیح است. باید خاطر نشان ساخت که رهنمود در واقع دی کامپایل را فقط محدود به هدف سازگاری ننموده است، بلکه دی کامپایل می تواند برای هدف تصحیح خطاها و اشکال زدایی برنامه هم طبق ماده (۱) ۵ نیز در نظر گرفته شود (Lehmann, 1993, p.178).

۴-۶. شروط مجاز بودن دی کامپایل

۴-۶-۱. شرط ضروری بودن دی کامپایل

مطابق ماده ۶ دی کامپایل باید «ضروری» برای کسب اطلاعات لازم برای سازگاری باشد و واجد شرایط مندرج در ماده ۱ باشد. یک نقطه مغفول در رهنمود، فقدان تعریف از مفهوم "ضروری" است. در حالی که برخی پیشنهاد دادند که مراد ضرورت های فنی است (Katoh, 1995, p.250)، اما باید پذیرفت که دادگاه های اروپایی باید با انعطافی معقول، اصطلاح مزبور را با توجه به رویه های موجود تفسیر نمایند. علت ذکر این شرط آن است که دی کامپایل آخرین بد ضروری باشد و البته رهنمود فرایند و ترتیبی برای بیان ضروری بودن مشخص نکرده و آن را در اختیار دادگاه ها قرار داده است. در عمل هم مهندسی معکوس به علت هزینه های بالا، زمان بر بودن و ناکارایی آن در بسیاری موارد به عنوان آخرین چاره اندیشیده می شود و نه اولین چاره.

۴-۶-۲. شرط مربوط به دی کامپایل کننده

شرط دیگر این است که دی کامپایل کننده، منحصراً یا شخص مجوز گیرنده و یا شخص مجاز دیگر باشد.

۴-۶-۳. شرط در دسترس نبودن اطلاعات

شرط دیگر این است که قبلاً به راحتی در دسترس نباشد. در ارتباط با شرط فوق، این سؤال مطرح می شود که آیا لازم است که برنامه نویس ابتدا درخواست کند که دارنده برنامه اصلی - به صورت رایگان یا با قیمت معقول - اطلاعات لازم را در اختیار او قرار دهد. این چنین شرطی توسط چند نهاد اروپایی بررسی و مردود اعلام شد.

منطق نهفته وجود شرط مزبور این است که پدیدآورنده برای جلوگیری کردن از دی کامپایل اثرش، تشویق شود اطلاعات را به شیوه آسان در اختیار دیگران قرار بدهد (Katoh, 1995, p.250).

برخی دیگر با چند مقدمه زیر در تأیید مطلب یادشده استدلال نموده‌اند (Sucker, 1993, pp.19-20):

۱. شرط یادشده این امکان را برای دارنده فراهم می‌کند که نسبت به دی کامپایل اثرش کنترل و نظارت داشته باشد چرا که با انتشار اطلاعات رابط کاربری برنامه اش، اشخاص ثالث از دی کامپایل اثر بدون رضایت او بازداشتت خواهند شد.

۲. چون این چنین اطلاعاتی باید به سهولت در دسترس باشد اشخاص ثالث چنانچه این اطلاعات در مستندات کلی و فنی برنامه گنجانده نشده باشد هیچ وظیفه‌ای برای درخواست نسبت به اینکه اطلاعات را برنامه نویس اصلی در اختیار قرار بدهد را ندارند.

۳. نمی‌توان ادعا نمود که اطلاعات به راحتی در دسترس است، چنانچه دارنده فقط در صورت پرداخت حق الزحمه و مابه‌ازاء، حاضر به در اختیار قرار دادن اطلاعات شود؛ زیرا این امر هدف دسترسی به اطلاعات را تضعیف می‌کند (Dreier, 1991, p.324).

۴-۶-۴. شرط محدود بودن دی کامپایل به بخش‌های لازم

شرط دیگر به موجب بند ۲ این ماده این است که دی کامپایل فقط محدود به بخش‌هایی می‌شود که برای سازگاری لازم است. البته تحقق این شرط در عمل بسیار سخت است؛ زیرا مشکل بلکه ناممکن است که محدودیت‌هایی را برای دی کامپایل کل برنامه ایجاد نمود، به ویژه اینکه مشکلات ذاتی مهندسی معکوس و هزینه بالای آن به گونه‌ای است که تکنیک‌های خاص مهندسی معکوس برای فراتر از اجزای ضروری برنامه در عمل استفاده نمی‌شود (Huet & Ginsburg, 1992, pp.363-364).

۴-۶-۵. شرط استفاده مجاز

دیگر محدودیت حق دی کامپایل مربوط به استفاده‌ای می‌شود که از نتایج دی

کامپایل بدست می‌آید. ماده ۶ استثنا را فقط برای هدف مربوط به سازگاری برنامه جدید مجاز دانسته است. برخی گفته‌اند که این چنین محدودیتی در ماده ۶ با نظام سنتی کپی‌رایت ناسازگار است؛ زیرا دسترسی به ایده‌های برنامه که ارتباطی با سازگاری ندارد، طبق نظام کپی‌رایت با توجه به تفکیک بین ایده و بیان آن اشکالی ندارد و این در حالیست که شرط مزبور سبب حمایت بی دلیل کپی‌رایت از ایده می‌شود (Katoh, 1995, p.253).

۴-۶-۶. سایر شروط

شروط دیگر استفاده این است که برنامه نویس نمی‌تواند هر اطلاعات اخذ شده از دی کامپایل را به دیگران بدهد مگر برای هدف سازگاری. شرط دیگر ماده ۶ این است که اطلاعات به دست آمده از طریق دی کامپایل نباید برای توسعه، تولید و بازاریابی برنامه‌ای که شباهت اساسی در بیان با برنامه دیگر دارد و یا برای هر عملی که نقض کپی‌رایت است به کار گرفته شود. عبارت بکار رفته در این ماده خیلی گسترده است؛ زیرا فراتر از کپی لفظ به لفظ کد می‌شود (Czarnota & Hart, 1994, p.81).

۴-۷. دی کامپایل فراتر از نیاز سازگاری

۴-۷-۱. نگهداری برنامه رایانه‌ای

نگهداری نرم‌افزار می‌تواند شامل موارد زیر باشد:

۱. ترجمه برنامه رایانه‌ای نوشته شده به زبان برنامه نویسی قدیمی و منسوخ شده به زبان برنامه نویسی جدید و مدرن.
۲. تبدیل کد برنامه از قالب‌ها و شکل‌های قدیمی (مثل برنامه نوشته شده با روش‌های کهنه شده و ابتدایی همچون روش اسپاگتی (Spaghetti Code) یا ساختار نیافته) به روش‌های جدید مثل برنامه نویسی با روش شی‌گرا (Object Oriented Programming (O.O.P)).
۳. بازنویسی کد برنامه‌ای که مفقود شده است.
۴. انتقال برنامه کاربردی از یک محیطی به محیط دیگر نرم‌افزاری یا سخت‌افزاری.

۵. گذشته از این، عمل دی کامپایل می‌تواند برای موضوعات امنیتی نیز مفید و با اهمیت باشد زیرا از طریق عمل دی کامپایل برنامه کاربردی همچون رمزگذاری نرم‌افزار یا سیستم‌های چک و کنترل مجوزهای دسترسی به دلایل امنیتی کاربرها قادر خواهند بود که این مسئله را چک کنند آیا نرم‌افزار شان خالی از ویروس است یا خیر (Walter, 2001, p.212).

این مسئله که آیا رهنمود این امکان را برای دی کامپایل برای مقاصد و اهداف فوق در نظر گرفته است یا خیر؟ در اینجا باید بررسی گردد. اگر چه عبارت پردازی و لحن ظاهری ماده ۶ به گونه‌ای است که اجازه دی کامپایل برای اهداف نگهداری نرم‌افزار را نمی‌دهد، اما با مراجعه به مستندات قانونگذار به خوبی معلوم می‌شود که قصد قانونگذار از متن قانون فراتر از معنای ظاهری آن است و شامل موارد فوق نیز می‌شود (Heindl, 2008, p.68). کمیسیون در نخستین پیش‌نویس خود از عبارت «ضروری و ناگزیر» استفاده کرد تا شامل موارد خلق، نگهداری یا ایجاد برنامه سازگار شود، اما به دلیل اینکه دی کامپایل عملی پرهزینه و پرمسئله است آن را فقط به اهداف سازگاری محدود نمود. از آنجایی که اکنون می‌توان برای مفهوم سازگاری مفهوم موسعی را قائل شد، بنابراین می‌توان ادعا نمود که این مفهوم وسیع حتی شامل فرض حفظ سازگاری برنامه رایانه‌ای در محیط‌های مختلف سخت افزاری و نرم‌افزاری و یا برنامه‌های رایانه‌ای جدیدتر می‌شود تا بسیاری از موارد یادشده را پوشش دهد، اما بر خلاف پیش‌نویس‌های ارائه شده به پارلمان، کمیسیون موضوع «حفظ و نگهداری» برای هدفی غیر از سازگاری را دلیل مستقلاً برای توجیه دی کامپایل تأیید نکرد، به ویژه اینکه به روزکردن یا پیشرفته کردن نسخه قبلی برنامه از وظایف انحصاری دارنده نرم‌افزار است نه لیسانس گیرنده و کاربر (Lehmann, 1993, p.78).

البته ممکن است به شیوه دیگر، جواز دی کامپایل به منظور نگهداری را از طریق ماده ۵ رهنمود استنباط نمود. در این ماده بیان شده است که تکثیر و ترجمه کد برنامه نیازمند اجازه دارنده نیست در جایی که این عمل برای استفاده کاربر مجاز از برنامه رایانه‌ای برای رسیدن به اهداف برنامه ضروری باشد. البته این که چه عملی را می‌توان اهداف خاص برنامه دانست، در رهنمود مشخص نشده است و مثلاً معلوم نیست آیا

هر عمل نگهداری به خودی خود شامل ماده ۵ میشود به شرطی که در راستای اهداف قصد شده در برنامه باشد. یکی از راه های شناسایی قصد و هدف لیسانس دهنده، مراجعه به متن قرارداد لیسانس یا استناد به اهداف کلی لیسانس در موارد مشابه است؛ بنابراین بدیهی است بند ۱ ماده ۵ هم تنها یک مبنای متزلزل برای توجیه مجاز نمودن دی کامپایل به منظور حفظ برنامه به حساب می آید. البته هنوز این سؤال باقی است که آیا تصحیح خطا هم در حوزه قلمرو حفظ برنامه واقع می شود یا خیر؟ باتوجه به هدف ذاتی برنامه که اجرای بدون ایراد و خطا است، به نظر پاسخ مثبت است (Marly, 1995, p.329).

۲-۷-۴. تصحیح خطاها

ممکن است کاربر برنامه به استناد بند ۱ ماده ۵ رهنمود ادعا نماید که تکثیر، ترجمه، اقتباس یا هر تغییر دیگر برای تحلیل خطاها و رفع آنها بدون اجازه از دارنده مجاز است به شرطی که تصحیح خطا برای استفاده از برنامه لازم باشد. فرض تصحیح خطاها در قانون بسیار ابهام آمیز است؛ زیرا نه آن تعریف شده است و نه از مفهوم خطا و باگ (= اشتباه نرم افزاری) تعریف مشخصی ارائه شده است؛ زیرا علیرغم معنای فنی خطا و اشتباه نرم افزاری، چه بسا کاربری یک عمل کامپیوتر را خطا بداند، اما کاربری دیگر همان را یک ویژگی خاص برنامه تلقی نماید. به علاوه اگر تصحیح خطاها توسط کاربر مجاز باشد، این بسیار برای برنامه نویس سودمند است بدون اینکه هزینه ای پرداخت کرده باشد که در این صورت مسئله سواری مجانی (Free Riding) برای او اتفاق می افتد (Spoor, 1994, p.169).

ذیل ماده ۵ رهنمود، به مانند ماده ۶ عمل نکرده است که شخص را ملزم کند اطلاعات کسب شده را در اختیار دیگران قرار ندهد و برای اهداف دیگری استفاده نکند، اما نظر می رسد همان شروط ماده ۶ باید برای ماده ۵ هم اعمال شود. همچنین ماده ۵ مشخص نکرده است که آیا قاعده ای امری است که با توافق قراردادی قابل نادیده گرفتن نباشد یا قاعده اختیاری است و به عنوان پیش فرضی است که در صورت تصریح به خلاف در قرارداد بر اساس شرط قراردادی باید عمل شود و بنابراین از این

جهت مبهم است و نتیجه این ابهام ارائه تفاسیر مختلفی است که در قوانین کشورهای عضو از آن شده است. به عنوان نمونه قانون پرتغال این قاعده را امری می‌داند. شیوه دیگری که در انگلستان اتخاذ شده است شرط ضمنی قراردادی جواز تصحیح خطاهاست^۱. (British Leyland Motor Corp. & Ors. v. Armstrong Patents Company Ltd. & Ors., 1986).

۴-۷-۳. هدف تحقیق و پژوهش

به مانند موارد قبل رهنمود صراحت ندارد در این که آیا دی کامپایل برنامه کامپیوتری برای هدف تحقیق فراتر از مفهوم سازگاری است یا خیر. برای روشن کردن قلمرو دقیق موارد شرایط سازگاری مجاز، نظام حقوقی امریکا و انگلیس راهنمایی‌هایی را نسبت به موضوع فوق ارائه داده است قبل از پیاده شدن مقررات رهنمود در انگلستان دکترین رفتار منصفانه که معیاری در امریکاست نسبت به اهداف تحقیق ارائه شد. باتوجه به فقدان مقرره‌ای درباره پژوهش در رهنمود ازیکسو و وجود مقرره با رفتار منصفانه در فرض پژوهش این مقرره اخیر جایگزین مقرره رهنمود شد (Marly, 1995, p.293).

۴-۷-۴. حمایت از دانش موجود در برنامه به عنوان نتایج قصد نشده از سازگاری

گرچه ماده یک رهنمود مقرر کرده که ایده‌ها و قواعدی که اجزای برنامه بر اساس آن بنا شده‌اند مشمول حمایت کپی‌رایت هستند. به علاوه شرط ماهوی اصالت اثر شرط اصلی برای حمایت در نظام کپی‌رایت است. لذا جنبه‌هایی از برنامه که شرط اصالت را ندارند از حمایت برخوردار نیستند. با وجود این ماده ۶ به نظر می‌رسد که تمامی این اصول را زیر پا گذاشته و دی کامپایل را برای هر هدفی به جز سازگاری ممنوع اعلام کرده است بدون توجه به اینکه آیا عناصر قابل حمایت در آن موجود است یا خیر؛ بنابراین توسعه آن به موارد غیر قابل حمایت به موجب کپی‌رایت تنها اغراق از اصول کلی کپی‌رایت است، بلکه شیوه منحصر به فرد و انحصاری در نظام کپی‌رایت اروپا

1. available at: <http://www.bailii.org/uk/case/UKHL/1986/7.html> (last visited at Sept. 18, 2018).

ایجاد می‌کند (Lehmann, 1993, p.23).

۵. مهندسی معکوس در نظام کپی‌رایت ایران

۵-۱. مهندسی معکوس نرم‌افزار در قوانین فعلی کپی‌رایت

با عنایت به نوپا بودن صنعت نرم‌افزار و موجود نبودن آن به هنگام وضع قانون کپی‌رایت ۱۳۴۸ ایران، پرواضح است که در این قانون به نرم‌افزار و احکام اختصاصی آن هیچ اشاره‌ای نشده باشد. به همین خاطر، قانونگذاران با تصویب قانون ویژه حمایت از پدیدآورندگان نرم‌افزارهای رایانه‌ای در سال ۱۳۷۹ در صدد رفع این خلاء قانونی برآمدند. مواد مرتبط با موضوع بحث در این قانون عمدتاً مواد ۱۵ و ۷ و مواد ۱۲، ۱۳ و ۱۵ آیین‌نامه قانون می‌باشد که مرتبط‌ترین آنها ماده ۵ قانون و ماده ۱۲ آیین‌نامه است. گرچه به اطلاق ماده ۱ این قانون، هرگونه تکثیر بدون کسب اجازه از پدیدآورنده هرچند در فرآیند دی‌کامپایل صورت گرفته باشد، نقض حقوق انحصاری پدیدآورنده است، اما در این ماده، قانون‌گذار با تعبیر بسیار مبهمی امکان خلق نرم‌افزارهای مکمل و سازگار را با رعایت حقوق مادی نرم‌افزارهای اصلی مجاز دانسته است. از آنجایی که پدیدآمدن نرم‌افزار سازگار در صورت در اختیار نداشتن مستندات فنی رابط‌های برنامه اصلی (اینترفیس برنامه) نیازمند دی‌کامپایل نرم‌افزار است، بنابراین لازمه اطلاق ماده ۵ جواز این امر به منظور ایجاد برنامه مکمل و سازگار خواهد بود، اما احتمال اینکه دی‌کامپایل مشمول ذیل ماده ۱۷ باشد بسیار بعید و دور از ذهن است. علاوه بر آن قانون تجارت الکترونیکی ۱۳۸۲ در ماده ۶۳ در خصوص فضای مجازی، تکثیر موقت در فضای مجازی را به شرطی که جزء لاینفک فنی پردازش باشد را مجاز اعلام نموده است و ممکن است با تنقیح مناط، تکثیر موقت در فرایند دی‌کامپایل نرم‌افزار در محیط عادی را نیز شامل شود.

۳۲۱

حقوق اسلامی / نقدی بر استنباط مهندسی معکوس نرم‌افزارها در تحولات قضایی...

۱. ذیل ماده ۷ تکثیر نرم‌افزاری که به طور مجاز برای استفاده شخصی به طور همزمان تهیه شده است را بلا مانع می‌داند.

۲-۵. مهندسی معکوس نرم افزار در لایحه قانون جامع حقوق مالکیت ادبی و هنری و حقوق مرتبط

این لایحه که هدف آن تجمیع مقررات مندرج در قوانین مختلف اصلی و یا مرتبط با کپی رایت ۱ در یک قانون واحد و به روز نمودن و بومی سازی آن متناسب با تحولات شتابان این زیر شاخه حقوقی است، اهمیت ویژه‌ای را برای مهندسی معکوس قائل شده و در فصل استثنائات حقوق مادی، عنوانی را به این موضوع اختصاص داده است. ۲. الگوی اصلی این قسمت، مواد ۶، ۵ و ۲۲ رهنمود ۱۹۹۱ برنامه‌های رایانه‌ای اتحادیه اروپا (اصلاح شده در سال ۲۰۰۹م) است که این امر تأییدی بر تلاش قانونگذاران بر به روز نمودن مقررات کپی رایت ایران در حوزه نرم افزارها است. با وجود این اقتباس پذیری و الهام گیری از مقررات مزبور، لایحه با ایرادات و اشکالات متعددی همراه است که در ادامه به آن می‌پردازیم، اما قبل از نقد این مواد ابتدا لازم است به اختصار مفاد مواد یادشده در لایحه تشریح گردند. متن ماده مزبور در لایحه اینچنین است:

«ماده ۲۲ - دستیابی به فناوری‌های به کاررفته در یک برنامه رایانه‌ای از طریق تحلیل ساختار، عملکرد و وظیفه آن (شالوده کاوی برنامه رایانه‌ای) برای ایجاد برنامه رایانه‌ای سازگار بدون کسب اجازه از دارنده حق مجاز است، به شرط آنکه: ۱- دسترسی به اطلاعات برای کار با برنامه رایانه‌ای سازگار به طور مستقل یا با برنامه‌ای دیگر ضرورت داشته باشد. ۲- برای هدفی غیر از به کار انداختن برنامه رایانه‌ای سازگار از آن استفاده نشود و جز برای نیل به این هدف در اختیار شخص دیگری قرار داده نشود. ۳- برنامه رایانه‌ای ایجادشده به صورت تجاری اجرا، تکثیر و مورد استفاده قرار نگیرد.

ماده ۲۳ - کاربر مجاز برنامه رایانه‌ای میتواند به منظور شناسایی ایده‌ها و اصول نهفته در ورای هر یک از عناصر برنامه، آن را مورد ملاحظه، مطالعه و آزمایش عملکرد قرار

۱. همچون قانون ۱۳۴۸ (قانون مادر: حمایت حقوق مؤلفان و مصنفان و هنرمندان)، قانون ۱۳۵۲ (ترجمه و تکثیر کتب و نشریات و آثار صوتی) و قانون ۱۳۷۹ (حمایت از حقوق پدیدآورندگان نرم افزارهای رایانه‌ای).

۲. این لایحه طی مواد ۲۲ تا ۲۳ به ابعاد مختلف مهندسی معکوس پرداخته است.

دهد»

عنوان به کار رفته در این قسمت به جای اصطلاح رایج مهندسی معکوس نرم افزار یا دی کامپایل، اصطلاح ترجمه شده «شالوده کاوی» است که علیرغم مزیت آن از جهت تلاش قانونگذار بر جایگزینی اصطلاحات فارسی به جای معادل خارجی آن یعنی دی کامپایل با این اشکال مواجه است که نمی تواند آینه تمام نمای اصطلاح دی کامپایل یا مهندسی معکوس نرم افزار در معنای عام خود باشد؛ زیرا نقطه ثقل کلمه «شالوده کاوی» تلاش جهت یافتن اسرار فنی نرم افزار است که این به خودی خود باتوجه به تفکیک بین ایده و بیان در نظام کپی رایت مجاز بوده و اصولاً در قلمرو حمایت این حوزه قرار ندارد و مربوط به زیر شاخه حقوق اسرار تجاری می شود. بلکه آنچه در اینجا مهم است دو فرایند تکثیر و اقتباس اثر دیگران است که در حالت عادی نقض کپی رایت دارنده اثر به حساب می آید که در فرآیند دی کد کردن یا دی کامپایل کد مقصد اتفاق می افتد و تکثیر و اقتباس از لوازم عمل مزبور است و این مفهوم به طور ضمنی از عبارت دی کامپایل استفاده می شود.

۳-۵. موضوع ماده

موضوع این ماده به مانند عنوان این قسمت از این ناحیه دچار اشکال است که «دستیابی به فناوری های به کار رفته در برنامه رایانه ای» عنوان گشته است، در حالی که همانطوریکه بیان شد صرف دستیابی به فناوری های به کار رفته در برنامه رایانه ای در قلمروی حمایتی حوزه کپی رایت قرار ندارد و آنچه که لازم است این ماده به آن تأکید نماید و به عنوان موضوع ماده قرار گیرد «تکثیر و اقتباس» صورت گرفته در طی انجام فرآیند دی کامپایل است؛ بنابراین طراحی ماده ۲۲ به شکل موجود به نظر نادرست و ناقص می آید و چون در صدد بیان استثنائات حقوق مادی است به نظر می رسد این چنین باید تعبیر شود: «تکثیر، تبدیل، ترجمه و اقتباس اثر در فرآیند دی کامپایل نرم افزار

«...»

۴-۵. اشخاص مجاز

از این منظر، نقد وارد بر ماده ۱۱ لایحه این است که در این ماده قانون گذار فهرست اشخاصی که حق شالوده کاوی یا دی کامپایل برنامه رایانه‌ای را دارند را بیان نکرده است، در حالی که شرط نخست این رهنمود اتحادیه اروپا جهت مجاز بودن دی کامپایل این است که عمل مزبور تنها توسط یکی از این سه دسته صورت گرفته باشد: «۱. انتقال گیرنده حق بهره برداری ۲. توسط هر شخص دیگری که حق استفاده از کپی برنامه را دارد ۳. توسط شخصی که به حساب آنها عمل می‌کند انجام گیرد» بنابراین به قانون گذار ایرانی توصیه می‌شود در این ماده فهرست اشخاص مجاز یادشده تصریح شود.

۵-۵. شرط در دسترس نبودن اطلاعات

یکی از شرایط مجاز بودن دی کامپایل طبق ماده ۶ رهنمود این است که اطلاعات قبلا به آسانی در اختیار شخص قرار نگرفته شده باشد. مع الاسف، در لایحه این شرط آورده نشده است. بدیهی است ذکر شرط فوق این فایده را دارد که پدیدآورنده نرم افزار اصلی تشویق به در اختیار عموم قرار دادن مستندات فنی رابط کاربری می‌شود تا دیگران اقدام به دی کامپایل برنامه (و در نتیجه افشای رموز و فنون آن) ننمایند. به دیگر سخن، دیگران نتوانند به بهانه دی کامپایل برنامه برای سازگار نمودن برنامه‌هایشان از فنون و رموز فنی برنامه مطلع شوند؛ بنابراین توصیه می‌شود که قانونگذار این شرط را به شروط جواز دی کامپایل اضافه نماید.

۶-۵. محدود نمودن دی کامپایل به بخش‌های ضروری برنامه

یکی دیگر از شروط دی کامپایل مجاز (مندرج در بند ج ماده ۶ رهنمود) محدود بودن آن به قسمت‌های ضروری است که برای سازگاری مفید می‌باشد، اما این شرط در لایحه مورد غفلت قانونگذار واقع شده است. تنها توجیهی که می‌توان برای حذف این شرط در لایحه بیان کرد این است که امکان تشخیص و جداسازی بخش‌هایی که برای دی کامپایل ضروری است عملا برای شخص دی کامپایل کننده فراهم نیست مگر بعد

از دی کامپایل کل برنامه. البته مشابه این نقد هم توسط برخی حقوقدانان اروپایی بر نظیر این ماده در رهنمود اتحادیه اروپا مطرح شده است (Vinje, 1992, p.251).

۷-۵. شرط عدم مشابهت برنامه سازگار با برنامه اصلی

طبق شرط سوم در ماده ۶، دی کامپایل به شرطی مجاز است که برنامه سازگار شباهت اساسی با برنامه پیشین نداشته باشد؛ زیرا این به معنای کپی غیر لفظی و در پاره‌ای موارد لفظ به لفظ کد برنامه اصلی خواهد بود و پر واضح است این امر نقض حق تکثیر دارنده برنامه اصلی است. با وجود این، شرط مزبور از دید قانونگذار ایرانی نادیده گرفته شده است. از این رو، اضافه شدن آن به شروط ماده ۲۲ به قانونگذار توصیه می‌شود.

۸-۵. فقدان ارائه تعریف دی کامپایل

دی کامپایل از جمله مفاهیمی است که احکام خاص حقوقی در لایحه بر آن بار شده است و این در حالی است که از لحاظ فنی نسبت به حدود و ثغور آن اختلاف وجود دارد و شفاف سازی مفهومی این اصطلاح و تعیین مرزهای آن در لایحه در بخش مفاهیم ضروری است که متاسفانه در آن قسمت لایحه نیز تعریف نشده است. به علاوه شالوده کاوی که در لایحه آمده، ترجمه مهندسی معکوس است نه دی کامپایل، در حالی که موضوع استثنائات، دی کامپایل نرم افزار که خاص تر از مهندسی معکوس است، می‌باشد.

۹-۵. محدود شدن به سازگاری

محدود کردن هدف دی کامپایل برای سازگاری مذکور در بند ۲ ماده ۲۲ نیز با قواعد کلی نظام کپی رایت ناسازگار است؛ زیرا چنانچه هدف از دی کامپایل کشف ایده‌های موجود در برنامه باشد، این امر طبق نظریه تفکیک ایده از بیان در نظام کپی رایت بلامانع است؛ بنابراین مجاز نبودن دی کامپایل به منظور کشف ایده‌ها خارج از قلمرو نظام سستی کپی رایت است و از این جهت موجه و منطقی به نظر نمی‌رسد. از طرف

دیگر بدیهی است دی کامپایل با هدف تست و اشکال زدایی برنامه از ملزومات لاینفک استفاده از برنامه رایانه‌ای است که قطعاً باید مجاز اعلام شود، در حالی که در لایحه دی کامپایل با هدف غیر سازگاری مطلقاً ممنوع دانسته شده است و همانطوریکه قبلاً گفته شد، در رهنمود اتحادیه اروپا جواز دی کامپایل را نه تنها با هدف سازگاری بلکه به منظور تصحیح خطاها نیز مجاز دانسته است (ماده ۵). تنها توجیهی که می‌توان ارائه داد این است که قانونگذار ایرانی در ماده ۲۱ امکان تکثیر یا اقتباس از برنامه را با هدف تصحیح خطاهای موجود در آن مجاز اعلام کرده است و ممکن است ادعا شود اطلاق آن شامل تکثیر در فرآیند دی کامپایل به منظور فوق نیز می‌شود.

۵-۱۰. عدم تصریح به امری بودن مقرر

باتوجه به اهمیت مستثنا شدن دی کامپایل نرم‌افزار، رهنمود آن را در زمره مقررات امری قرار داده است به طوری که نمی‌توان با شروط قراردادی آن را محدود کرد. با وجود این مقنن ایرانی در اقتباس از رهنمود، در هیچ ماده‌ای از لایحه نسبت به عدم امکان محدودسازی دی کامپایل از طریق شروط قراردادی اشاره‌ای نکرده است.

۵-۱۱. دی کامپایل با هدف پژوهش

یکی از مواردی که لازم است در لایحه جزو اهداف مجاز دی کامپایل آورده شود علاوه بر هدف سازگاری، هدف پژوهش و تحقیق است که متأسفانه در لایحه آورده نشده است و این در حالی است که در نظام حقوقی غرب با ابزارهایی همچون نظریه استفاده منصفانه این امر موجه دانسته شده است.

نتیجه

دی کامپایل برنامه یا تبدیل کد مقصد یا کد ماشین غیر قابل فهم به کد مبدا قابل فهم برای انسان یکی از استثنائات حقوق مادی است که در کشورهای اروپایی تحت تاثیر رهنمود برنامه‌های رایانه‌ای ۱۹۹۱ اتحادیه اروپا درباره برنامه‌های رایانه‌ای مجاز اعلام شده است؛ زیرا در جریان دی کامپایل برنامه عمل تکثیر ترجمه و اقتباس از اثر

اصلی انجام می‌پذیرد. این امر گرچه به حسب اصول کلی نقض حقوق مادی پدیدآورنده باید تلقی شود، اما در صنعت نرم‌افزار، ضرورت‌های ساخت نرم‌افزارهای سازگار و مکمل توسط دیگران قانونگذار کشورهای پیشرو را مجبور ساخته تا به جهت رعایت منافع عمومی و تشویق مردم به خلق و افزایش و توسعه آثار نرم‌افزاری این امر را به شرط اینکه با هدف انجام برنامه سازگار باشد مجاز اعلام نمایند. لایحه قانون حمایت از مالکیت ادبی و هنری و حقوق مرتبط گرچه از این امتیاز برخوردار است که به پیروی از ماده ۶ رهنمود اتحادیه اروپا دی‌کامپایل را تحت شرایطی مجاز اعلام نماید، اما به دلیل عدم تبعیت کامل از اتحادیه اروپا بسیاری از شروطی که در این اتحادیه ذکر شده است را نیاورده است. در این نوشتار کلیه شروط پیش گفته بررسی و خلأهای قانونی لایحه با مطالعه تطبیقی از این منظر بررسی گردید و اضافه شدن بسیاری از شروط اتحادیه اروپا در لایحه پیشنهاد شد و نسبت به عدم ذکر پاره‌ای از شروط توجیهاتی نیز ارائه گردید.

منابع

۱. قانون ترجمه و تکثیر کتب، نشریات و آثار صوتی ۱۳۵۲.
۲. قانون حمایت از حقوق پدیدآورندگان نرم‌افزارهای رایانه‌ای ۱۳۷۹.
۳. قانون حمایت حقوق مؤلفان، مصنفان و هنرمندان ۱۳۴۸.
۴. لایحه حمایت از مالکیت ادبی و هنری و حقوق مرتبط ۱۳۹۳ (آخرین اصلاحات بهار ۱۳۹۸).
5. Anthony J. Mahajan, Intellectual Property, Contracts, and Reverse Engineering After ProCD: A Proposed Compromise for Computer Software, 67 Fordham L. Rev. 3297 (1999).
6. Band, Katoh, Interfaces on Trial (Westview Press, 1995).
7. Blythe, Stephen E. "The U. S. Digital Millennium Copyright Act and the E. U. Copyright Directive: Comparative Impact on Fair Use Rights" 8 Tul. J. Tech. & Intell. Prop. 111 (2006)
8. Boris Rotenberg, the Legal Regulation of Software Interoperability in the EU, NY Jean Monnet Working Paper No. 07/2005.
9. Brian C. Behrens & Reuven R. Levary, Legal Aspects - Software Reverse Engineering and Copyright: Past, Present and Future, 31, The John Marshall Law Review, 1 (1997)
10. Bridget Czarnota, Robert J. Hart, Legal Protection of Computer Programs in Europe: A Guide to the EC Directive, Rutgers Computer and Technology Law Journal, C. XX (1994).
11. British Leyland Motor Corp. & Ors. v. Armstrong Patents Company Ltd. & Ors. (1986) 2 W. L. R. 400 (Feb. 27, 1986) (U. K.).
12. Carver, Brian W. "Share and Share Alike: Understanding and

- Enforcing Open Source and Free Software Licenses” 20
Berkeley Tech. L. J. 443 (2005)
13. Celine M. Guillou, the Reverse Engineering of Computer Software in Europe and the United States: A Comparative Approach, 22 COLUM. -VLA J. L. & ARTS (1998).
 14. Council Directive of 14 May 1991 on the Legal Protection of Computer Programs, 1991.
 15. Derek Prestin, “Where to Draw The Line Between Reverse Engineering and Infringement: Sony Computer Entertainment, Inc. v. Connectix Corp. ”, 3 Minn. Intell. Prop. Rev. 137, 173 (2002).
 16. Dreier “The Council Directive of 14 May 1991 on the Legal Protection of Computer Programs” [1991] 9 EIPR 319.
 17. Germany Copyright Law 1998
 18. Goldstein, Paul. Goldstein on Copyright, New York, NY: Aspen Publishers, 2007 (3rd Ed.)
 19. Herald Jongen D. J. , & Alfred p.Meijboom, (Eds.). Copyright Software Protection in the EC, Deventer-Boston: Kluwer Law and Taxation Publishers, 1993.
 20. Jaap H. Spoor, Copyright Protection and Reverse Engineering of Software: Implementation and Effects of the EC Directive, 19 U. DAYTON L. REV. 1063, 1065-6 (1994).
 21. Jerome Huet and Jane Ginsburg “Computer Programs in Europe: a Comparative Analysis of the 1991 EC Software Directive” (1992).
 22. Jin-Keun Jeong, Comparative Study on Legal Protection in the

- USA, EU, Japan and Korea for Computer Programs —Focus on Program Reverse Engineering-, IIP Bulletin , Vol. 17, 2006
23. John V. Howard, Jr. , Jill Gushow, Interoperability in the United States and Europe, (1995)
 24. Johnson-Laird, A. (1994) “Software reverse engineering in the real world” University of Dayton Law Review, vol. 19
 25. Lee, Donna L. “Reverse Engineering of Computer Programs under the DMCA: Recognizing a ‘Fair Access’ Defense” 10 Marq. Intell. Prop. L. Rev. 537 (2006)
 26. Lehmann, Handbook of European Software Law (1993) p.178 (“The European Directive on the Protection of Computer Programs”).
 27. Marly, Urheberrechtsschutz für Computersoftware in der Europäischen Union [Copyright Protection for Computer Software in the European Union] (1995).
 28. Michael Sucker, Handbook of European Software Law (1993).
 29. MICHEL M. WALTER, EUROPÄISCHES URHEBERRECHT [European Copyright Law] (2001).
 30. Pamela Samuelson & Suzanne Scotchmer, the Law and Economics of Reverse Engineering, 111 Yale L. J. (2002).
 31. 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?, STANFORD – VIENNA TRANSATLANTIC TECHNOLOGY LAW FORUM (TTLF) WORKING PAPERS NO. 1 (2008)
 32. Rotenberg, Boris. “The Legal Regulation of Software

Interoperability in the EU” NY Jean Monnet Working Paper No. 07/2005 (2005).

33. Sega Enterprises, Ltd. v. Accolade Corp. , 977 F. 2d 1510 (9th Cir. 1993)
34. Sony Computer Entertainment, Inc. v. Connectix Corp. , 203 F. 3d 596 (9th Cir. 2000) , cert. denied, 531 U. S. 871 (2000).
35. Stanley Lai, the Copyright Protection of Computer Software in the United Kingdom, Oxford: Hart Publishing, 2000.
36. Sugiyama, “Copyright and Patent of Software”, Nippon Hyoronsha, 1999
37. Thomas C. Vinje, Die EG-Richtlinie zum Schutz von Computerprogrammen und die Frage der Interoperabilität [The EC Directive on the Protection of Computer Programs and the Question of Interoperability], 4 GRUR INT. 250, 259 (1992).
38. United States v. Microsoft, 65 F. Supp. 2d 1, 15 (D. D. C. 1999).
39. Walter, Michel M. (Ed.). Europäisches Urheberrecht, Wien: Springer Verlag, (2001).

۳۴