DRM And Allowing Authentication

Don Armstrong For Committee D

May 13, 2006

1 GPLv3 Clause

1. Source Code.

   [...] Complete Corresponding Source Code also includes any encryption or authorization codes necessary to install and/or execute the source code of the work, perhaps modified by you, in the recommended or principal context of use, such that its functioning in all circumstances is identical to that of the work, except as altered by your modifications. It also includes any decryption codes necessary to access or unseal the work's output. Notwithstanding this, a code need not be included in cases where use of the work normally implies the user already has it.

3. Digital Restrictions Management.

   As a free software license, this License intrinsically disfavors technical attempts to restrict users' freedom to copy, modify, and share copyrighted works. [...] no permission is given [...] for modes of distribution that deny users that run covered works the full exercise of the legal rights granted by this License.

2 Issue

This particular clause of the GPL has generated a rather large number of comments; the comment thread of the comments is that the DRM restriction as currently written appears to require that authentication keys, encryption keys, API keys, and any passphrases required to get them to function be provided.

■ **DLA:** *There are a few other issues tied in with this section; user data encryption will be addressed in the DRM and Freedom 0 issue, since I think it may be more applicable there.*

We will analyze a few of the key comments, and then make a series of suggestions which, if enacted, we hope will resolve these concerns.

2.1 Authentication Verification

> Currently, the DRM restriction as written also appears to dissallow the use of mechanisms to verify the authenticity of a piece of GPLed software which can be either extended to contain the end users keys in a documented manner, or totally disabled. [For example, this appears to make Debian's current method of assuring that users can authenticate the contents of our archives a violation of the GPLv3, which is a consequence that I believe is unintentional.] – Comment by don [450]

> [...] 2) Developers should: be able to sign software in a way that only allows it to be run in unmodified form AND still be identified as signed by the developer [...] 6) GPLed software should: be able to identify signed software packages and verify the origin of said software 7) GPLed software should NOT: be designed to stop users from *choosing* to run non-signed or unofficial code/modules [...] 6 & 7) RedHat uses signatures to verify that RPM packages were created by RedHat before installing. Users are (should be) given the choice to install non-RedHat RPMs if they choose to, but this may (negatively) affect their support contract with RedHat, which is perfectly acceptable. [...] – Comment by [798]

Beyond their use in treacherous computing, which (if we can be so bold to say) these clauses were written to address, signed packages are used to allow users to verify that a particular package came from a source that they trust and has not been tampered with since it was signed by the source that they trust.

So long as the user can override the authentication/verification system or install their own keys to make the work behave identically, it should not be necessary for the distributor to include keys used to identify the work as originating from the distributor.

2.2 API keys

In addition to authentication keys and passphrases, various API keys may be required to get a work to behave identically to the original version which the distributor possesses.

> This has potential impact on people writing code which uses webservices. Some websites have keys which must be provided when using the webservice. Under this term, a user using Google's API (for example), would be forced to include their private Google API key to comply with the GPL. – Comment by fraggle [370]

API keys are often used to uniquely identify a particular individual for interaction with a remote system, as occurs in credit card merchant systems and, as the comment indicates, the Google SOAP API.[1] Since these keys are necessary for the work to behave identically to the version that the distributor has, it appears that the distributor would need to distribute the API keys which they are using as well, a consequence which was likely unintentional. Indeed, the "Notwithstanding [...] a code need not be included [...]" sentence appears to have been added specifically to address this issue.

■ **DLA:** *I think we may need to expand the above section slightly, but I haven't thought about it completely yet.*

[1] The SOAP API is used to perform Google searches without needing to parse the HTML output to recover results.

2.3 Encryption Keys

There is also some cocern about the requirement of this section to provide private keys necessary to decrypt output when the code in question is designed to create encrypted output specific to a set of user-provided public and private keys[2].

> This [implies] that code for asymetric encryption of data (e.g. gnupg email encryption) cannot be put under v3, since the means for decryption of the output is the private key of the receiver, which is not available to the sender. [...] It might well be that "codes" is implying "algorithms" instead of "keys" here, which would solve the problem. – Comment by [661]

Like the API key issue, the "Notwistanding [...] a code need not be included [...]" also appears to address this, but there would be some cases where a user would not necessarily have the code to decrypt the data. Some clarification of the above sections will be useful in order to resolve the encryption key issue as well.

■ **DLA:** *I need to work through this part slightly more to figure out what I want to say about clarifying this section.*

2.4 Execute Source Code

A final trivial issue to address is "execute the source code" which doesn't make much sense outside of the scope of interpreted languages/compilers like elisp or perl. In our recommendation, we will operate under the assumption that the user was intended to have the ability to install, compile (if required) and execute the work (or a transformation of the work).

3 Committee Recommendations

Based on these comments we would like to suggest that one of the following changes be adopted to address these sections.

[2] That is, it is possible to encrypt the data without being able to decrypt the data.

■ **DLA:** *I was thinking originally about having people give specific assent to a set of these solutions, but I'm no longer convinced that RMS will actually care who in the committee agrees with a particular statement.*

3.1 Original

1. Source Code. Complete Corresponding Source Code also includes any encryption or authorization codes necessary to install and/or execute the source code of the work, perhaps modified by you, in the recommended or principal context of use, such that its functioning in all circumstances is identical to that of the work, except as altered by your modifications. It also includes any decryption codes necessary to access or unseal the work's output. Notwithstanding this, a code need not be included in cases where use of the work normally implies the user already has it.

3.2 Modification

1. Source Code. Complete Corresponding Source Code also includes any encryption or authorization codes necessary to install, compile (if required), and execute the work, perhaps modified by you, such that its functioning in all circumstances is identical to that of the work, except as altered by your modifications. It also includes any decryption codes necessary to access or unseal the work's output unless the user has specifically instructed the work to provide encrypted output, and can instruct it to provide unencrypted output. Encryption or authorization codes that the user can easily disable or replace with their own codes and retain identical functioning need not be provided, so long as the user can easily generate the codes or obtain them at no cost. A code need not be included in cases where the use of the work normally implies the user already has it.

■ **DLA:** *The above is still a bit suboptimal, unfortunatly. I'm really trying to say that you need to be able to either automatically generate your own codes and have them work in exactly the same manner, or you should be able to get your own code for free, or for at least no more money than you needed to pay for the source code.*

An argument can be made that the clause already does what we've asked it to change to, but I think the changes made above definetly clarify the clause.

3.3 WDiff

> Complete Corresponding Source Code also includes any encryption or authorization codes necessary to [- *install and/or* -] [+ **install, compile (if required), and** +] execute the [- *source code of the* -] work, perhaps modified by you, [- *in the recommended or principal context of use,* -] such that its functioning in all circumstances is identical to that of the work, except as altered by your modifications. It also includes any decryption codes necessary to access or unseal the work's [+ **output unless the user has specifically instructed the work to provide encrypted output, and can instruct it to provide unencrypted** +] output. [- *Notwithstanding this, a* -] [+ **Encryption or authorization codes that the user can easily disable or replace with their own codes and retain identical functioning need not be provided, so long as the user can easily generate the codes or obtain them at no cost. A** +] code need not be included in cases where [+ **the** +] use of the work normally implies the user already has it.

3. Digital Restrictions Management.

   As a free software license, this License intrinsically disfavors technical attempts to restrict users' freedom to copy, modify, and share copyrighted works. [...] no permission is given [...] for modes of distribution that deny

users that run covered works the full exercise of the legal rights granted by this License.

- **DLA:** *source code isn't executed*
- **DLA:** *user overridable or replaceable authentication keys need not be provided; perhaps making it more obvious that "necessary" authorization or encryption codes does not include codes which can be replaced by the end user. How to do this without gutting this clause entirely (ie, requiring keys that cost many thousands of dollars) is not yet clear.*

4 Replies to specific comments

> Consider the situation where I have generated a document that I would like to prevent from being copied from the machine. For the sake of the argument it's a confidential tender document for my business. It's copyrighted but my reasons for using DRM to secure it relate not to the copyright but for business reasons. Consider that Linux has somehow been relicensed to GPL v3 - am I now required to use a Windows workstation to control access to my data? – Comment by Unknown [602]

This section of the GPLv3 does not attempt to control the ability of users to control their own data on their own machines. It merely disallows the attempts of licensors to control the ability of users of the software (and licensees) to make and deploy modifications to the software by the use of cryptographic keys which the user cannot override.

5 Comments related but not included specifically

The following comments are related, but they have not been included specifically in this issue summary. It is possible that they will be included in the future. They will likely be removed from the final version before that version is handed to RMS.

The implications of the DRM clauses need to be carefully considered. While I disagree with DRM in its current form and see it as a problem better dealt with administratively than technically I can see where DRM can be put to good use. For example, I may want to restrict certain actions on a confidential business document - this is a less nefarious use of DRM than preventing me from exercising my fair use rights by copying a song to my media player. A more interesting example is where that business document is a pre-release version of a movie ('screener') - raises some interesting questions. – Comment by Unknown [594]

Like others, I find this confusing. Consider a keypad digital lock on a safe used to store backup tapes, or a door to a datacenter. Should I be able to unlock the door if I'm not supposed to go in? Or consider Crypto whose entire purpose is to protect data. If I send you a message encrypted with your key, I haven't given everyone else permission to read it. The difficulty is properly defining what are letters, sealed in envelopes designed for specific recipients (and if it is to my priest, lawyer, or doctor, I really, really, might not want the information distributed), and a book designed for anyone who purchases it to read. In the former case, I really want strong locks with GPL so I can review it to insure there are no backdoors, but in the latter case there ought to be no locks. The question might be one of private v.s. public information (even paid for or "members only" type). – Comment by Unknown [575]

Change this to "other software or hardware systems capable of accessing the same data" (so that mod-chips and such are allowed). – Comment by Unknown [536]

Digital Rights Management (DRM) need not involve restrictions. Beware conflating the common acronym DRM with yours. E.g. An operating system might provide users no ability to copy files, and only DRM "rights" software grants that ability. Such a non-GPL OS restricts by omission; GPL'd DRM software may **enable** limited extra uses. Arguably, that form of DRM may not involve extra restrictions, in fact it bypasses OS restrictions in certain, limited ways (perhaps subject to copyright holder permission). Your word "Restrictions" makes it unclear whether such DRM software can be GPL'd because it may limit this section's scope to only software which creates a restriction. – Comment by Unknown [531]

DRM must be defined. Laws and the general public understand it as a mean to control the user, which is clearly in contradiction with the purpose of the GPL. However, some people claim that it is mainly access control and information about the user rights. – Comment by Unknown [365]

I am concerned that these paragraphs are much too vague. If one favours freedoms, the words are possible to read as preventing DRM. But it is equally possible to read into these paragraphs permission for content encryption, making a loop-hole for the disingenuous. How is it that the GNU Privacy Guard, or any similar crypto application, is *not* such a disfavored program? How is it that any *use* of GPG is not considered a disfavored activity? After all, DRM technologies are crypto technologies at their most fundamental level. – Comment by Unknown [338]

Is this too specific? Imagine an embedded device where, in order to replace the firmware, you had to pass a particular complex pattern over an optical sensor. Is that an "authorisation code"? How about "any information or data necessary"? – Comment by Unknown [192]

There's a negative aspect to CCSC when I use the requirements of it upon GPG to force its authors to supply me with their private keys because they are "decryption codes necessary to access or unseal the work's output" when used by these fine people. Correspondingly, a bank who uses GPLv3 software would be required to supply the private keys used to sign their work if such a work is an object code executable downloaded to a computer or service accessed by the internet. I hope this doesn't come across as troll-like – I hope that this section can be clarified so that either my misunderstanding (should it be so) is corrected or this supposed loophole closed. – Comment by Unknown [633]

Section 3 is written in such a convoluted way that I do not know how it is to be commented. So I am commenting this paragraph instead. An important issue is whether DRM can be implemented as free software. The issue is double: - will the legislator make it legal to implement DRM with publicly available source code. - what free software licences will be compatible with such an implementation (typically BSD or GPLv2 are). Of course, it would be best not to have DRM, but that is not necessarily for the programmer to decide. Regarding the second point, as drafted in this paragraph, I do not know what is the meaning of: "from which escape is forbidden" 1. If the source code is provided, one can escape by modifying the program. But is this considered an available escape by the the GPLv3, since many people cannot program ? 2. What if modifying the program to escape is permitted by law ? 3. What if it is not, but one can still do it ? (the text does not say legal escape) 4. What if modification is permitted by law, but not dissemination of the modified program that escapes the DRM ? 5. What if law permits anything, but the computer has TCPA like features that prevent running non certified code (one could escape by getting another computer, or disabling the hardware check with a disruptor (soon to be invented)) 6. What if all computers on the market are TCPA controlled ? Remember that there are illegal programs, and that any GPL program can be modified/derived into an illegal one. Are this provision and section 3 of the licence compatible with clause 6 of the Open Source Definition. – Comment by [650]

In Europe, some groups have suggested the idea that free software could be used to implement DRM protections, in order to avoid the effect of laws that could prohibit developing any free software capable of reading "protected" (rather, restricted-use) software. Have the interactions of GPLv3 with that strategy been thoroughly examined? – Comment by [651]

Consider changing "all circumstances" to "all circumstances and for all potential users". Otherwise, a vendor may provide a license code that's user- specific or machine-specific. The current "all circumstances" may be construed to mean all possible ways to install and/or execute the program *by the direct recipient*. – Comment by [658]

This sentence seems to imply that code for asymetric encryption of data (e.g. gnupg email encryption) cannot be put under v3, since the means for decryption of the output is the private key of the receiver, which is not available to the sender. Can this be cleared up? It might well be that "codes" is implying "algorithms" instead of "keys" here, which would solve the problem. – Comment by [661]

This is unreasonable. There a reasonable ethical argument for requiring that code changes (derivative works) be made available under the same terms. There is no ethical argument for requiring that devices be sold with an implied or explicit contract allowing the user to modify the core software of the device. In other words, it's OK to require the source code to include scripts, but it is NOT OK to require that the user be able to install the software on a particular device. – Comment by [680]

I have published a detailed analysis of DRM with a concluding section on this section provisions. It is on-line in French and English at http://www.debatpublic.net/Members/paigrain/blogue/DRMS It gives a positive appraisal of the sole first part of second paragraph, and critical arguments on the rest of the section. – Comment by [690]

This may be troublesome. There are concerns about a script that encrypts a message with a public key and sends it to the person with the private key. Does the GPLv3 force distributors to provide the private key along with the script? The letter of GPLv3draft1 seems to mean this... I hope it's unintentional: it looks like a bug in the license phrasing. – Comment by [688]

This will be problematic according to laws e.g. in Germany. Example: BPjM is an institution which provides a web content filter list which has to be applied e.g. by schools (the list e.g. contains child porn sites). It is illegal to provide the list in plaintext which is fully understandable and acceptable. Now assume that there would be an open source application for e.g. squid that would handle this filtering. Such an application cannot be GPLv3 compliant without breaking the law. Try it another way. Let's say, the list data is SHA1 hashed. There's no decryption for hashes. I wonder if the use of a hash function is thus GPLv3 compliant. The same problem with signed sources. One would need to provide the private key thus voiding the whole purpose of signing. And signing may not be there to restrict the recipient but to assert that no tampering has taken place. The whole key stuff seems either not to be well thought through or it seems to have been created wit a very U.S. centric view. I doubt that without clearly narrowing changes the GPLv3 will be applicable in quite some countries as it causes violations of local law in cases where the law is not in place for DRM support but for good reasons. – Comment by [797]

I think most people agree, the DRM section of the GPLv3 draft needs some work. I've come up with a list of considerations that need to be reviewed while updating the DRM section of the GPL. I do think there needs to be a DRM section, I just think the current version over steps the bound of an Open Source software license. 1) Developers should: be able to use any DRM technologies in software which they have written (in entirety) themselves 2) Developers should: be able to sign software in a way that only allows it to be run in unmodified form AND still be identified as signed by the developer 3) Developers should NOT: be able to prevent/force users or hardware manufacturer's into DRM technologies, unless it is directly affecting the developer's software 4) Users should: be able to modify and run software that is GPLed 5) Users should NOT: be able to modify and redistribute software in any manner that permits claiming to be the "original" unmodified version 6) GPLed software should: be able to identify signed software packages and verify the origin of said software 7) GPLed software should NOT: be designed to stop users from *choosing* to run non-signed or unofficial code/modules 8) Hardware manufacturers should: be able to build and ship hardware that has DRM technologies 9) Hardware manufacturers should NOT: be able to get legislation passed that requires all hardware to be DRMed 10) Governments should: be able to use DRM-like technologies in government situations, but only when said technologies do not prevent future open-and-fair-bidding of vendors 11) Governments should NOT: be able to *require* the use of DRM in private sector situations examples: 1 through 5) I write a program and license it under GPL. The software is free to modify, but I have designed the system so that only my unmodified (signed) version can access regularly updated content that I also produce and charge for. I don't want modified versions of the software accessing my data stream. This does not prevent users from from modifying the software and using to access a different stream, it only protects content that I created. (Note that I can not imagine a scenario where this example would be technically possible without a higher level of DRM, such as at the OS or hardware level. It is simply an example.) 6 & 7) RedHat uses signatures to verify that RPM packages were created by RedHat before installing. Users are (should be) given the choice to install non-RedHat RPMs if they choose to, but this may (negatively) affect their support contract with RedHat, which is perfectly acceptable. 8& 9) If IBM wants to ship DRM hardware, and users

I believe that this will substantially limit the spread and acceptance of free software within the Government. Many Government networks, particularly those dealing with security and safety of life, require that software be standardized. This usually means that the software must be signed to be installed. Many of us have worked for a long time to have free software accepted and used in Government systems because we believe that — in the long run — Government use and acceptance will benefit both the free software movement and the public at large. Provisions such as this, and the digital rights restriction below, threaten to undermine these years of effort. Non- free software companies (e.g. Microsoft) can say "You can sign our software to insure it isn't changed." Because most Government users already believe that "You can't go wrong buying Microsoft" restrictions such as this may sound the death-knell of our years of advocacy. Lest you think this is hyperbole, let me tell you of my experience. I wanted to use free software on a secure government system. To do so required that I test the software thoroughly and then insure that it was distributed unmodified. It took me months to get this far and I had to leave the project before I could complete the process. But at least I make the security people consider the problem so that the next proposer may find it easier. Finally, as a licensed (registered) professional engineer, this restriction (combined with the DRM passages below) will prevent **me** from using free software in **any** project dealing with the safety of life or property. I **cannot** allow the **possibility** that critical software might be changed. To do so would violate my professional responsibility because it would put the safety of my clients — and the general public — at risk. Perhaps I misunderstand this passage or the DRM passage below. I hope so. Because if they really mean what I believe they mean, they will defend the theory of free software while perversly discouraging its practical use. – Comment by [817]

Please see my long comment above. Safety or security-related software **must** be standard. This may **require** digital signing or other DRM-like restrictions. This passage will perversely discourage the use of free software in those applications.
authorization codes necessary to install and/or execute the source code of the work, except as altered by your modifications. Your phrasing allows this interpretation, i.e. "your modifications" need not be unlocked. Is this intended? – Comment by [753]

Some countries have adopted laws prohibiting software that enables users to escape from Digital Restrictions Management. DRM is fundamentally incompatible with the purpose of the GPL, which is to protect users' freedom; therefore, the GPL ensures that the software it covers will neither be subject to, nor subject other works to, digital restrictions from which escape is forbidden in any one country. – Comment by [752]