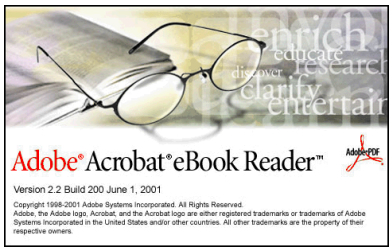
Global Information Assurance Certification Paper

ELCOMSOFT vs. ADOBE
How AEBPR cracked Adobe's Acrobat E-Book Reader



On July 16th 2001, Russian cryptologist Dmitry Sklyarov was arrested after delivering a presentation at the Def Con hackers' conference in Las Vegas. He was charged with distributing a product designed to circumvent copyright protection measures under the Digital Millennium Copyright Act (DMCA). Behind the arrest was a claim filed by Adobe Systems Incorporated that the Russian software company ElcomSoft was distributing a product removing the security restrictions from Adobe's Acrobat E-Book Reader. The Advanced E-Book Processor (AEBPR), initially available on the Elcomsoft website for $99,[1] was purported to override any restrictions placed on an e-book by a publisher, and convert the document into a standard, unprotected Adobe Acrobat Reader PDF format.[2] The story has captured worldwide media attention, and many debates discussing the political, legal, and philosophical ramifications of the issue have resulted.[3]

While some believe that products such as AEBPR open the door for widespread piracy of copyrighted e-book material, others believe that companies such as Elcomsoft are working in the interests of e-book publishers: by exposing inherent weaknesses in any security model, holes are able to be patched, security improved, and material ultimately better protected from piracy. Dmitry Sklyarov's cause has been taken up by civil liberties groups such as the Electronic Freedom Foundation (EFF); protest actions have been staged and websites such as www.freesklyarov.org and www.boycottadobe.com have emerged. At the time of writing (August 2001), Dmitry Sklyarov was released on bail pending trial in Northern California; even though Adobe has dropped out of the Federal government's case against Sklyarov, the government is continuing to pursue prosecution of the case.

Central to the debate is Adobe's Acrobat E-Book Reader security model, its use of encryption and plug-ins known as security handlers. This paper will look at this security model, and examine how a program such as AEPBR has been able to circumvent it.

---

[1] The software has since been removed from the website http://www.elcomsoft.com/aebpr.html
[2] These claims have been substantiated by Adobe Certified Expert Bryan Guignard in his whitepaper, and Roger Sperberg in his two part EbookWeb article. Both were able to remove the security restrictions on E-Book Reader documents using AEBPR.
[3] For a complete list of articles and web discussions see the Planet eBook Index at http://www.planetebook.com/mainpage.asp?webpageid=170

E-Book Basics

An e-book or electronic book is a digital book that you can read on a personal computer, on a handheld device such as a Palm or Pocket PC, or on a specialized e-book reading device such as Gemstar's eBook (formerly RocketBook and SoftBook).[4] To read an e-book on a personal computer, you must have a piece of software called an E-Book Reader. The key players in this field are Adobe's Acrobat E-Book Reader (formerly GlassBook Reader) and the Microsoft Reader. Both of these products can be downloaded at no cost from many sites on the Internet, including the major online book retailers Amazon.com and BarnesandNoble.com. The Microsoft Reader can be used on any Windows platform, including the Pocket PC, whilst Adobe's Acrobat E-Book Reader is compatible with both Windows and Macintosh platforms.

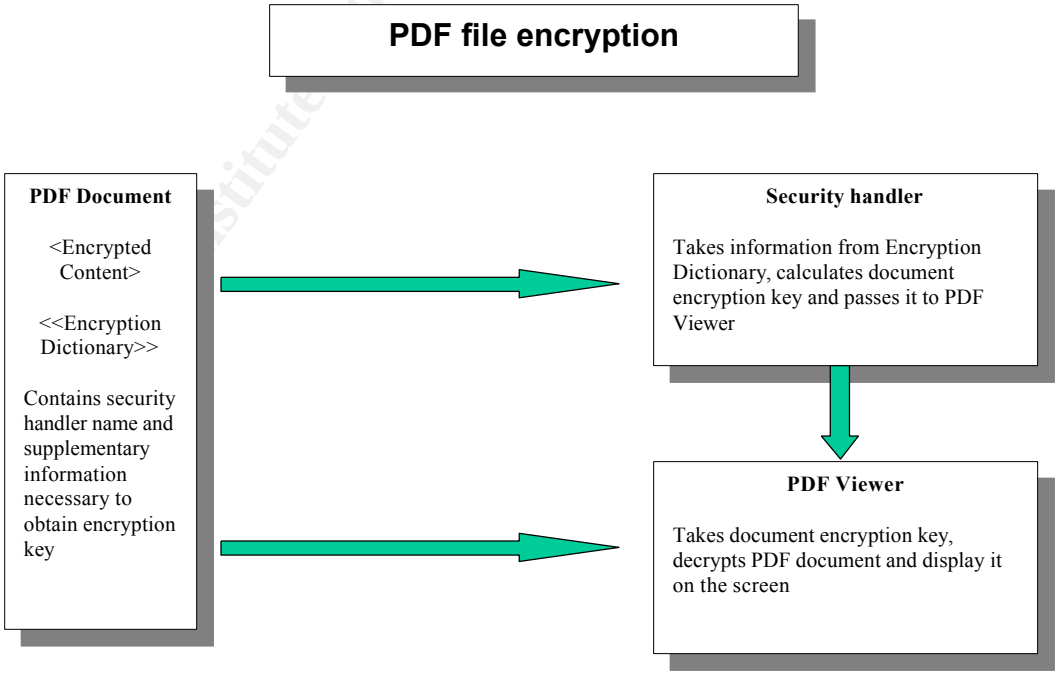
Before a user can purchase an e-book via the Internet, they must download and register their copy of the Reader software online. For e-book publishing to become a commercially viable field, e-book authors, publishers, and retailers must be assured not only that online monetary transactions will be secure, but also that their content will be safe from unauthorized modification and distribution once purchased. Potentially, just one legally purchased copy of an e-book could be duplicated into countless pirated copies if security is not adequately implemented. As Adobe states: "While catalyzing new business opportunities, this electronic delivery model also of the raises complex questions about the protection of digital rights. Publishers, distributors, and resellers have learned a lesson from watching the music industry struggle with the consequences of digital music distribution."[5]

The process of securing copyrighted digital content is encompassed by the field of Digital Rights Management (DRM). Digital Rights Management (DRM) is defined as "the technologies, tools and processes that protect intellectual property during digital content commerce, [and] is a vital building block of the emerging electronic book (e-book) market." [6] Adobe's Acrobat E-Book Reader and its web-based Content Server (the server which handlers e-book distribution for publishers) supports a number of integrated and third party DRM systems such as PDF Merchant, EBX, SoftLock and FileOpen.

[4] Information about e-book hardware devices can be found at http://www.ebookad.com/hardware.php3
[5] Adobe System Inc., September 1999. p 1.
[6] http://www.w3.org/2000/12/drm-ws/pp/macgrawhill-bolick.html

E-Book Reader Security Model

Like the Adobe Acrobat Reader, the Acrobat E-Book Reader is a Portable Document Format (PDF) viewing application. PDF files are platform-independent in that they may be viewed by a PDF viewing application on any supported platform, regardless of the application, software, or operating system they were created on. Derived from postscript language, a PDF file is a collection of objects combined with some additional informational objects. Each object is uniquely identified by a combination of object number and generation number. A PDF document can be protected from unauthorized access by encrypting the string and stream objects of the document to protect them from being read by unauthorized parties. It can also be protected by applying permissions that allow or deny a user access to various functions within a document such as copying or printing. The permissions that can be selected for an e-book (set by the publisher via the Adobe Content Server) are the ability to copy, print, give, or lend a document, and the option of reading out loud. All encrypted PDF documents will contain an encryption dictionary – this is a file that contains information on the permissions that are to be applied, the type of encryption that is to be used, and the name of the security handler that will control the security of the document. The function of a security handler is to calculate the key used for encryption, and maintain the values in the encryption dictionary. The diagram below illustrates the roles that the security handler and PDF viewer play in PDF encryption.[7]



[7] Diagram adapted from Dmitry Sklyarov's Def Con presentation, p 7.

E-Book Reader Encryption

The standard PDF encryption algorithm is RC4, a proprietary algorithm of RSA Data Security Incorporated. RC4 is a symmetric algorithm in that the same key is used for both encrypting and decrypting data. This means that all parties requiring access to the data must hold a copy of the same key. Obviously in symmetric encryption systems, the secrecy of the key is of paramount importance as data can be decrypted by any unauthorized parties gaining access to the key. The RC4 algorithm is a stream cipher in that it generates cipher text (that is, text that has been encrypted) using a sequence of bits used as a key known as a "keystream". Encryption is accomplished by combining the plain unencrypted text of a document (in ASCII format) with the sequence of bits used in the keystream. The two sets of bits are combined using the XOR process. This means that when the two data streams are compared bit by bit, the XOR process returns '0'if two bits are the same, and '1' if two bits are different. When used in combination with relatively large key lengths, this cipher can be very secure.[8]

RC4 is an algorithm that allows for variable key lengths depending on which version of the algorithm is used. Version 1 allows for 40-bit lengths only, whilst version 2 and 3 allow for lengths up to 64 and 128-bits. When using key lengths of 40-bits, it is possible to determine the encryption key by trying every possible combination (brute force), although this may take some time and require a great deal of computer power. Some companies on the Internet offer to break encryption on PDF files based on 40-bit key lengths. For example Password Crackers Inc. will search for the key used on an encrypted Acrobat file for a fee of $500. They state "there are fewer keys than passwords, hence we are able to search for all possible keys in less than 25 days."[9] In his Def Con presentation, Dmitry Sklyarov calculated that one PIII-450 computer would take an average of 40 days per document to crack one 40 bit-key. However by increasing the number of computers and the amount of memory used, this time could be significantly decreased.[10] The e-book industry has already seen the limitations of using 40-bit key lengths. When author Stephen King released Riding the Bullet in July 2000, pirated copies of the book were circulating the Internet within days with claims that encryption had been broken.[11] In an effort to deter cracking by brute force, greater key lengths are encouraged. With a larger number of bits, the number of possible keys is greatly increased making cracking much more difficult. As Adobe states:

> With 64-bit encryption, there are 20 billion billion possible keys to decipher the coded information, and only one of them works. Someone intercepting the

[8] Information about the RC4 algorithm is posted on the RSA website http://www.rsasecurity.com/rsalabs/faq/2-1-5.html
[9] Password Crackers Inc http://www.pwcrack.com/pdf.htm
[10] Dmitry Sklyarov, Def Con Presentation, p10.
[11] Information on the Stephen King e-book crack can be found at http://www.cnn.com/2000/books/news/03/30/king.pirated/

> information would have to find the right key — a nearly impossible task. With 128-bit encryption, the number of possible keys is the square of the number of 64-bit keys. It is virtually impossible for an unauthorized party to find the right key, even if that party is equipped with the best computers.[12]

While this may be true, the developers of AEBPR contend that increased key lengths do not necessarily increase security. As the customer is provided with a copy of the key when they purchase an e-book, they simply have to locate the key that is stored somewhere on their computer. AEBPR will help them to do this. Although the location of the key may differ depending on the security handler used, AEBPR will be able to locate it and resave the document into a plain, unprotected PDF format. The following is a statement from the ElcomSoft website:

> We claim that ANY eBook protection, based on Acrobat PDF format (as Adobe eBook Reader is), is ABSOLUTELY insecure just due to the nature of this format and encryption system developed by Adobe. The general rule is: if one can open a particular PDF file or eBook on his computer (does not matter with what kind of permissions/restrictions), he can remove that protection (by converting that file into "plain," unprotected PDF). Not very much experience [is] needed. In brief: ANY security plug-in…does nothing but returns a decryption key to Adobe Acrobat Reader or Adobe Acrobat eBook Reader. Plug-ins can make various hardware verifications, use parallel port dongles, connect to the publisher's web site and use asymmetric encryption, etc, but all ends up with a decryption key, because the Reader needs it to open the files. And when the key is there, we can use it to decrypt the document removing all permissions.[13]

In his presentation to Def Con, Dmitry Sklyarov outlined how AEBPR removes the security restrictions of six security handlers: the PDF Standard Security Handler, Rot13, FileOpen, SoftLock, PDF Merchant, and EBX. Let's look at the PDF Standard Security Handler in more detail to see how this is done.

E-Book Reader Crack

All security handlers are responsible for computing a document key that can be used to encrypt and subsequently decrypt a document. Each security handler can calculate this key in a different way. The PDF Standard Security Handler calculates the encryption key by using two password strings known as the "owner password" and the "user password." These passwords are usually randomly generated. Each password is padded or truncated to 32 bytes and placed within a file's encryption dictionary. The 32 bytes of each password, the file's permissions in binary form (contained within the encryption dictionary), and a unique file identifier are input into the Message Digest 5 (MD5) hash

[12] Posted on Adobe's website http://www.adobe.com:80/products/contentserver/overview2.html
[13] Previously posted at http://www.elcomsoft.com/aebpr.html

function. A hash function is a method for transforming data in such a way that it cannot be changed back to its original form. Hash functions are often used in e-commerce to verify the integrity of a file; that is has not been modified in transit, and the authenticity of a file; that it was sent by the person claiming to have sent it. They are also used to send sensitive information such as passwords, across a network. MD5 is a one-way mathematical algorithm that can take any length of data and produce a 128-bit "fingerprint." This fingerprint is "non-reversible" in that it is not possible to determine a file's contents based on its fingerprint alone.

Depending of the length of the key required (as specified in the encryption dictionary) the first 40, 64, or 128-bits of the MD5 output are used as the document key. This key is then used to encrypt the contents of the document using the RC4 algorithm as discussed earlier. When a customer purchases an e-book, it is a copy of this key, along with the encrypted document that is downloaded. The customer is also supplied with a valid user password that is used by the document key to decrypt the e-book and display its contents on the E-Book Reader screen. Using the user password, a customer will only be allowed to view the contents of the e-book and carry out those functions specified by the publisher. To remove all of these restrictions and resave the document in an unencrypted format, the owner password must be supplied. [14]

AEBPR is able to crack documents that use the PDF Standard Security Handler if either a valid user or owner password is known. Once supplied with a valid password, the document key will be called and begin to decrypt the document. As the document key will always pass through the MD5 hash function in the process of decrypting a file, AEBPR is able to intercept the document key by intercepting the hash function. According to Sklyarov, locating the MD5 hash function is not difficult. The MD5 hash function is usually called just after the MD5_init function, which always uses the same constants as defined in the MD5 specifications.[15] Once the key has been intercepted it can be used to resave the document into a standard PDF format, minus any of the original permissions defined by the publisher. The author is then free to print, copy and distribute the document as they see fit.

ElcomSoft maintains that the AEBPR program will only work on documents where a legitimate user password is supplied, indicating that a copy of the document has been legally purchased. Some argue that a customer should be able to have as many copies of a legally purchased document as they wish. They may wish to create backup copies, or view the document on another computer or platform not yet supported by the E-Book Reader. For example, there is currently no Reader software that supports a Linux platform. This is a legal concept known as fair use rights - the ability to use a part of a copyrighted work for a recognized legitimate purpose, without having to seek the prior permission of the copyright holder.[16]

---

[14] Detailed information about PDF Standard Security Handler encryption is available in the "Portable Document Format Reference Manual Version1.3" at http://partners.adobe.com/

[15] MD5 specifications are found in RFC 1321 at http://www.cis.ohio-state.edu/rfc/rfc1321.txt

[16] A good discussion of the legal implications of the Sklyarov case, including the concept of fair use is found

As stated earlier, all of the security handlers supported by the E-Book Reader manage security in a different way with some adding additional layers of security to the standard PDF security model. For example PDF Merchant[17] and EBX[18] increase security by encrypting the document key itself before downloading it to the customer. They use a voucher or certificate, in the form of a separate XML document, to verify that a customer is who they claim to be. Once the voucher has been verified, the encrypted document key (encrypted by the publishers "public key") is passed to the customer and must be decrypted by the customer's unique "private key" before the e-book can be opened. This is a form of asymmetric encryption in that two complimentary keys are used – although the keys are different, they are mathematically related allowing the private key to decrypt content encrypted by the public key. It is important to note however, that despite these added layers of security, AEBPR is still able to override any restrictions placed on an e-book using these security handlers. As with the PDF Standard Security Handler, the weakness lies in the fact along with any additional vouchers, certificates or asymmetric keys, a document key is always passed to the viewer, and as a result, there will always be one place on the customer's computer where the key is available. According to Elcomsoft, as long as a key is provided to the Reader and stored locally, products like AEBPR will be able to override protections and e-book solution developers will be unable to promote their programs as secure. If ElcomSoft is correct, the security model currently employed by Adobe needs to be revisited. Regardless of whether or not AEBPR is on the market, other products will eventually emerge that take advantage of the weaknesses in this model. The judicial system may decide the fate of Dmitry Sklyarov, but the battle between the e-book industry and its would-be pirates will be ongoing.

---

on the EFF website: http://www.eff.org/IP/DMCA/US_v_Sklyarov/us_v_sklyarov_faq.html

[17] PDF Merchant specifications are detailed in the "Adobe PDF Merchant SDK Reference Manual" at http://partners.adobe.com/

[18] EBX is the protocol of the Electronic Book Exchange Working Group. Its specifications are publicly documented on it's website http://xml.coverpages.org/ebx.html

References

Adobe Systems Inc. "Portable Document Format Reference Manual Version 1.3." 11 March 1999.

Adobe Systems Inc. "Adobe and Digital Content for eCommerce." September 1999.

Adobe Systems Inc. "Adobe PDF Merchant SDK Reference Manual." 2 December 1999.

Adobe Systems Inc. "Portable Document Format: Changes from Version 1.3 to 1.4." 11 June 2001.

De Abrew, Karl. "Using Adobe's PDF Merchant for Secure E-Book Distribution." URL: http://www.planetpdf.com/mainpage.asp?webpageid=884&nl

Electronic Book Exchange Working Group. "Electronic Book Exchange System (EBX) Version 0.8." July 2000. URL: http://xml.coverpages.org/ebx.html

Guinard, Bryan. "How Secure is PDF?" Whitepaper July 2001 URL: http://www.ebookweb.org/opinion/roger.sperberg.20010715.aebpr.htm

Merz, Thomas. "E-Commerce-Selling PDFs over the Web." PDFlib GmbH, Munchen. URL: www.pbflib.com

Sklyarov, Dmitry. "eBooks security – theory and practice" Def Con 9 Presentation, 13-15 July 2001. URL: http://www.planetebook.com/mainpage.asp?webpageid=170

Sperberg, Roger. "Removing Those Pesky Passwords." 12 July 2001. URL: http://www.ebookweb.org/opinion/roger.sperberg.20010712.aebpr.htm

Sperberg, Roger. "The Adobe Security Imbroglio." 15 July 2001. URL: http://www.ebookweb.org/opinion/roger.sperberg.20010715.aebpr.htm

Spitzner, Lance. "What is MD5 and why do I care?" URL: http://www.enteract.com/~lspitz/md5.html

Wizaerd's Forum. "PDF Security not so secure after all." Wizaerdsrealm Discussion, August 2001. URL: http://www.wizaerdsrealm.com/forum/