

1 DAVID W. SHAPIRO (NYSBN 2054054)  
United States Attorney

2 LESLIE CALDWELL (NYSBN 1950591)  
3 Chief, Criminal Division

4 SCOTT H. FREWING (CSBN 191311)  
JOSEPH SULLIVAN (FLSBN 988723)  
5 Assistant United States Attorneys

6 280 South First Street, Suite 371  
San Jose, California 95113  
7 Telephone: (408) 535-5060

8 Attorneys for Plaintiff

OT 12/13/01  
DEC 13 '01  
U.S. DISTRICT COURT  
NO. 01-20138

10 UNITED STATES DISTRICT COURT  
11 NORTHERN DISTRICT OF CALIFORNIA  
12 SAN JOSE DIVISION

14 UNITED STATES OF AMERICA,

15 Plaintiff,

16 v.

17 DMITRY SKLYAROV,

18 Defendant.

No. CR 01-20138 RMW

PRETRIAL DIVERSION AGREEMENT.

19 I, Dmitry Sklyarov, and the United States Attorney's Office for the Northern District of  
20 California (hereafter "the government") enter into this written Pretrial Diversion Agreement (the  
21 "Agreement"):

22 Background

23 1. Dmitry Sklyarov has been charged with committing offenses against the United States  
24 on dates unknown, but beginning no later than June 20, 2001 and continuing until July 15, 2001, in  
25 violation of Title 18, United States Code, Section 371 (conspiracy to traffic in a technology primarily  
26 designed to circumvent, and marketed for use in circumventing technology that protects a right of  
27 a copyright owner); Title 17, United States Code, Section 1201(b)(1)(A) (trafficking in technology  
28

PRETRIAL DIVERSION AGREEMENT  
CR 01-20138 RMW

1 primarily designed to circumvent technology that protects a right of a copyright owner); and Title  
2 17, United States Code, Section 1201(b)(1)(C) (trafficking in a technology marketed for use in  
3 circumventing a technology that protects a right of a copyright owner). It appearing, after an  
4 investigation of the offense and the defendant's background, that the interest of the United States,  
5 the interest of the defendant, and the interest of justice will be best served by the following  
6 procedure, the parties enter into this Agreement.

7 Defendant's Promises

8 2. I agree and certify that I am Dmitry Sklyarov, the above-named defendant. I agree  
9 the following facts are true:

10 A. Beginning on a date prior to June 20, 2001, and continuing through July  
11 15, 2001, I was employed by the Russian software company, Elcomsoft Co. Ltd. (also known as  
12 Elcom Ltd.) (hereinafter "Elcomsoft") as a computer programmer and cryptanalyst.

13 B. Prior to June 20, 2001, I was aware Adobe Systems, Inc. ("Adobe") was  
14 a software company in the United States. I was also aware Adobe was the creator of the Adobe  
15 Portable Document Format ("PDF"), a computer file format for the publication and distribution of  
16 electronic documents. Prior to June 20, 2001, I knew Adobe distributed a program titled the Adobe  
17 Acrobat eBook Reader that provided technology for the reading of documents in an electronic format  
18 on personal computers. Prior to June 20, 2001, I was aware that documents distributed in the Adobe  
19 Acrobat eBook Reader format are PDF files and that specifications of PDF allow for limiting of  
20 certain operations, such as opening, editing, printing, or annotating.

21 C. Prior to June 20, 2001, as a part of my dissertation work and as part of my  
22 employment with Elcomsoft, I wrote a part of computer program titled the Advanced eBook  
23 Processor ("AEBPR"). I developed AEBPR as a practical application of my research for my  
24 dissertation and in order to demonstrate weaknesses in protection methods of PDF files. The only  
25 use of the AEBPR is to create an unprotected copy of an electronic document. Once a PDF file is  
26 decrypted with the AEBPR, a copy is no longer protected by encryption. This is all the AEBPR  
27 program does.

28 D. Prior to June 20, 2001, I believed that ElcomSoft planned to post the AEBPR

1 program on the Internet on the company's website www.elcomsoft.com. I believed that the company  
2 would charge a fee for a license for the full version of the AEBPR that would allow access to all  
3 capabilities of the program.

4 E. After Adobe released a new version of the Adobe Acrobat eBook Reader that  
5 prevented the initial version of the AEBPR program from removing the limitations or restrictions  
6 on an e-book, I wrote software revisions for a new version of the AEBPR program. The new version  
7 again decrypted the e-document to which it was applied. The version of this new AEBPR program  
8 offered on the Elcomsoft website only decrypted a portion of an e-document to which it was applied,  
9 unless the user had already purchased a fully functional version of the earlier version and had both  
10 versions installed on the same machine. The new version was developed after June 29, 2001. At that  
11 time, Elcomsoft had already stopped selling the program. The version of this new program offered  
12 on the Elcomsoft website did not provide a user with an opportunity to purchase it or convert it to  
13 a fully functional one, and was developed as a matter of competition.

14 F. On July 15, 2001, as part of my employment with Elcomsoft, I attended  
15 the DEF CON Nine conference in Las Vegas, Nevada. At the conference I made a presentation  
16 originally intended for the BlackHat conference that immediately preceded the DefCon Nine in July  
17 2001 in Las Vegas, Nevada. The same group of people organizes both BlackHat and DefCon Nine.  
18 Since there was no available slot for a presentation at BlackHat at the time when the paper was sent  
19 for the committee consideration, the organizers of both conferences suggested that the paper be  
20 presented at the DefCon rather than at BlackHat. The paper that I read at DefCon is attached as  
21 Exhibit A. A principal part of my presentation is comprised of my research for the dissertation. In  
22 my presentation when I said "we", I meant Elcomsoft.

23 3. I assert and certify that I am aware of the fact that the Sixth Amendment to the  
24 Constitution of the United States provides that in all criminal prosecutions the accused shall enjoy  
25 the right to a speedy and public trial. I am also aware that Rule 48(b) of the Federal Rules of  
26 Criminal Procedure provides that the Court may dismiss an indictment, information, or complaint  
27 for unnecessary delay in presenting a charge to the Grand Jury, filing an information, or in bringing  
28 a defendant to trial.

1           4.       I hereby request that the United States Attorney for the Northern District of California  
2 defer any prosecution of me for the violations set out in paragraph one (1) of this Agreement for the  
3 stated period of months, and to induce him to defer such prosecution I agree and consent that any  
4 delay from the date of this Agreement to the date of the continued pursuit of the prosecution, as  
5 provided for in the terms expressed herein, shall be deemed to be a necessary delay at my request.  
6 I agree that this period of delay should be excluded from computation under the Speedy Trial Act  
7 pursuant to 18 U.S.C. § 3161(h)(2). I further waive any defense to such prosecution on the ground  
8 that such delay operated to deny my rights to a speedy trial under Rule 48(b) of the Federal Rules  
9 of Criminal Procedure and the Sixth Amendment to the Constitution of the United States, or that  
10 prosecution was barred by reason of the running of the statute of limitations for the period of months  
11 which is the period of this Agreement.

12           5.       I agree to abide by all the conditions and requirements of this Agreement, and I agree  
13 not to commit or attempt to commit any United States federal, state, or local crimes during the  
14 pendency of this Agreement. I also agree not to violate the terms of my pretrial release as stated in  
15 the Order dated August 6, 2001, or as modified by the Court; or intentionally provide false  
16 information to the Court, Pretrial Services, or the government.

17           6.       I agree to appear willingly and to respond truthfully to all questions put to me at: (1)  
18 a deposition on December 13-14, 2001; (2) at trial in *United States v. Elcom Ltd., a/k/a Elcomsoft*  
19 *Co. Ltd.*, CR 01-20138; (3) at hearings in *United States v. Elcom Ltd., a/k/a Elcomsoft Co. Ltd.*, CR  
20 01-20138, if subpoenaed through my undersigned U.S. counsel who retains the right to move to  
21 quash on all relevant grounds; and (4) at a further deposition pursuant to F.R.Crim.P. 15 if ordered  
22 by the Court in *United States v. Elcom Ltd., a/k/a Elcomsoft Co. Ltd.*, CR 01-20138. In addition,  
23 I agree to respond to questions about the AEPBR or its development put to me in writing through  
24 my counsel, or orally by means of a telephone interview with my counsel on the line. I agree this  
25 agreement constitutes consent to a deposition within the meaning of F.R.Crim.P. 15(d)(1). I agree  
26 that any statements made by me during any interview, deposition or trial testimony can be used  
27 against me in a prosecution for perjury, false statement, or obstruction of justice.

28           7.       I agree that should I violate the conditions of this program, the United States Attorney

1 may revoke or modify any of the conditions of this program or change the period of supervision,  
 2 which shall in no case exceed twelve (12) months or until the conclusion of the District Court  
 3 proceedings in the matter of *United States v. Elcom Ltd., a/k/a Elcomsoft Co. Ltd.*, CR 01-20138  
 4 RMW, whichever period is longer. I agree that the United States Attorney may release me from  
 5 supervision at any time, and that the United States Attorney may at any time within the period of my  
 6 supervision initiate prosecution for this offense should I violate any of the conditions, furnishing me  
 7 with notice specifying the conditions of the program which I have violated.

8 8. I agree that if I fail to comply with any promises I have made in this Agreement, that  
 9 the government will be released from its promises, but that my agreement that the facts described  
 10 above in paragraph two (2) are true and any subsequent statements made by me pursuant to this  
 11 Agreement may be used by the government against me in any proceeding, and I waive any and all  
 12 claims under the United States Constitution, Rule 11(e)(6) of the Federal Rules of Criminal  
 13 Procedure, Rule 410 of the Federal Rules of Evidence, or any other federal statute or rule, to  
 14 suppress or restrict the use of my statements, or any leads derived from those statements.

15 The Government's Promises

16 9. On the authority of the Attorney General of the United States, by David W. Shapiro,  
 17 United States Attorney for the Northern District of California, prosecution in this District for the  
 18 above offense shall be deferred for the period of twelve (12) months from this date or until the  
 19 conclusion of the District Court proceedings in the matter of *United States v. Elcom Ltd., a/k/a*  
 20 *Elcomsoft Co. Ltd.*, CR 01-20138 RMW, whichever period is longer, provided the defendant, Dmitry  
 21 Sklyarov, abides by the following conditions and requirements:

22 CONDITIONS OF PRE-TRIAL DIVERSION

23 A. The defendant shall not violate any U.S. law: federal, state, or local. The  
 24 defendant shall immediately contact his pretrial services officer if he is arrested and/or questioned  
 25 by any U.S. law enforcement officer.

26 B. The defendant shall report to his pretrial services officer as directed and keep  
 27 him/her informed of the defendant's whereabouts.

28 C. The defendant shall associate only with law-abiding persons.

D. The defendant shall follow the following special conditions:

(1) The defendant shall report in person or if in Russia via telephone once a month to United States Pretrial Services, and submit a written monthly report as directed;

(2) The defendant shall submit to interviews and a deposition as provided in paragraph six (6) of this Agreement; and

(3) The defendant shall make himself available pursuant to paragraph six (6) above for testimony at any hearing, or at trial, in the matter of *United States v. Elcom Ltd., a/k/a Elcomsoft Co. Ltd.*, CR 01-20138 RMW. The defendant shall not be responsible for a violation of this condition as a result of the government's failure or delay in approving any necessary visa or other travel procedure, provided the defendant promptly applies for such approval or procedure.

10. The government agrees that following the defendant completing the December 13-14, 2001, deposition described in paragraph six (6), the government will recommend to the Court that the defendant's pretrial release conditions be modified to allow him to reside in Russia.

11. If, upon completion of the defendant's period of supervision, a pre-trial diversion report from the defendant's pretrial services officer states to the effect that the defendant complied with all the conditions mentioned above, no prosecution for the offenses set out on page one (1) of this Agreement will be instituted in this District, and any indictment or information will be discharged.

Defendant's Affirmations

12. I hereby state and affirm that this Agreement has been read and explained to me in Russian. I understand the conditions of my pre-trial diversion and agree that I will comply with them.

DATED: 12/13, 2001

DAVID W. SHAPIRO  
United States Attorney

  
SCOTT H. FREWING  
Assistant United States Attorney

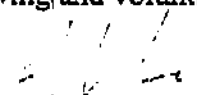
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

DATE: 12/13, 2001

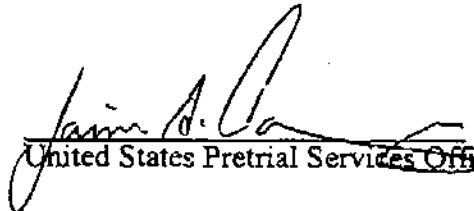
  
\_\_\_\_\_  
DMITRY SKLYAROV  
Defendant

I have fully explained to my client all the rights that a criminal defendant has and all the terms of this Agreement. In my opinion, my client understands all the terms of this Agreement and all the rights he is giving up by entering into this Agreement, and, based on the information now known to me, his decision to enter this Agreement is knowing and voluntary.

DATE: 12/13/2001

  
\_\_\_\_\_  
JOHN KEKER, ESQ.  
Attorney for Dmitry Sklyarov

DATE: 12/13/2001

  
\_\_\_\_\_  
United States Pretrial Services Officer

INTERPRETER CERTIFICATION

I, L. B. Chernov hereby certify that I am a certified Russian interpreter and that I accurately translated this plea agreement to the defendant.

DATE: 12/13/01

  
\_\_\_\_\_  
Interpreter's Signature

## eBooks security – theory and practice

Speech on DEF CON Nine, July 13<sup>th</sup> - 15<sup>th</sup>, 2001  
Alexis Park in Las Vegas, Nevada USA

### Foreword

My name is Dmitry Sklyarov. I'm employee of the ElcomSoft Company. We're developing security-related software for Windows platform.

I wish to start my speech from several words about goals of this presentation. Initial information about flaws in eBooks security was obtained during my researches for dissertation work that dedicated to estimating degree of security for different electronic publishing solutions. After collecting additional facts, we were just terrified. There are many solutions on the market, which positioned as secure, but they are not. And any publisher or distributor puts itself in a big danger by choosing secure eBook technology when guided only by information, obtained from that technology developer.

We have a long thought about way to share our knowledge about security flaws with others. We could make it public in all details, but such approach will, at first, strike to publishers, which are not responsible for developer's mistakes or facts obscurity. We could make demonstration program with limited functionality, but it is often practice for developers (especially for big one) – do not take in consideration any unwanted fact, until that fact becomes really threatening. Unfortunately, demonstration version is not threatening at all due to its demonstration, but not action nature.

Finally, we have decided to create demonstration program, which could be converted to the fully functional version by paying some amount of money. We have chosen \$100 as sort of threshold value. That value is several times larger than average eBook cost. So, not too many people will risk spending such amount. But developers will be unable to push ahead insecure solutions as secure at the fact of existing of such program.

It is a story of Advanced eBook Processor creation. The only thing Advanced eBook Processor does, is getting input file in PDF or eBookPro format and saving it without any protection. By the way, Advanced eBook Processor has ability to decrypt books in Adobe's Acrobat eBook Reader format, which actually just PDF with custom security handler.

Five days after initial release of the Advanced eBook Processor, ElcomSoft becomes a subject of attack from Adobe Systems Inc. Adobe claimed that Advanced eBook Processor is illegal and needed to be removed immediately from the web.

A day later, Internet hosting provider Verio Inc. has closed access to the ElcomSoft's web site (and several other sites with the same IP address) according to complains from Adobe. One of the reasons provided was that site "offers downloads to copyrighted software published by Adobe Systems", which is obviously not true.

On the same day Adobe sent complaint to RegNow, billing service used by ElcomSoft. This time they called it "unauthorized distribution of software".

Meanwhile Adobe has released version 2.2 of the eBook Reader software, which prevents eBooks from being deprotected by the first version of Advanced eBook Processor. But the only thing Adobe has changed in security mechanism was encryption key value. 000046 EXHIBIT A



Now ElcomSoft's web site is hosted with another provider, and demonstration version of Advanced eBook Processor, which could easily process books from eBook Reader 2.2, available.

You could find full story of conflict at <http://www.elcomsoft.com/aebpr.html>

But now, let's get back to technical aspects of the eBooks Security.

As we have demonstrated in our speech on Black Hat Win2K Security (February 2001), encryption in Microsoft Office documents is very weak and password protection may be removed without any problems in most cases. In this speech I'll try to cover some security aspects of electronic books and documents. The most attention will be paid to documents in PDF format.

Documents publishing in electronic form have a lot of advantages against traditional on-paper publishing. You could easily find list of such advantages on web server of any company, which provides eBook solutions. But nobody perfects, and there is one big problem that related with eBooks. Information in electronic form could be duplicated and transmitted, and there is no reliable way to take control over that processes. There are several solutions from different companies that were developed to prevent unauthorized distribution of the electronic documents.

Some solutions are oriented on dedicated devices like RocketBook by Gemstar. Such devices have ability to buy and download secured books over Internet. But in this speech I'm dealing with software-only solutions, which works on personal computers platform.

There are multiple programs that take HTML pages and pack them in container. After that container transferred to customer's computer and displayed there via Internet Explorer APL. Content of such books could be easily extracted from Explorer. Besides, developers of that software often have insufficient knowledge in security area. For example, The Internet Marketing Center, developer of eBook Pro software (<http://www.ebookpro.com>) proudly describes their product with following phrase: "eBook Pro", the only software in the universe that makes your information virtually 100% burglarproof! But on practice all data protection is concluded in XOR-ing each byte of compressed data with the same constant!

On my opinion there are only two popular formats, which allows relatively secured eBooks distribution. These formats are Microsoft LIT and Adobe PDF. Lets lefts Microsoft Reader with its LIT format for another speech, and start speaking about PDF.

First of all, I will talk about internal structure of PDF documents, which was mainly derived from PostScript language. Any document is set of objects plus some additional information, which defines special objects such as document root object, document information object, document encryption object, etc., and allows random access to any object in file.

Each object in document identified by unique combination of the object ID and generation ID. There are five basic data types could be used within object. They are Boolean values, numeric values, object references, names, strings, and streams. Streams are usually contains pages content in compressed form. Also two complex data types are supported: arrays and dictionaries. Dictionary is list of named elements of any type.

### *PDF encryption*

Now lets pay some attention to general principles of the PDF encryption. Documents can be encrypted to protect their contents from unauthorized access. Access to a protected document's contents is controlled by the security handler specified in the Encryption dictionary of document.

Strings and streams in a protected document, except those in the Encryption dictionary, are encrypted using the RC4 encryption algorithm (a copywritten, proprietary algorithm of RSA Data Security, Inc.). This prevents unauthorized users from simply removing the password from a PDF file to gain access to it. Other data types (such as integers and Booleans) that are used primarily for structural information in a PDF file are not encrypted. This combination protects a document's contents, while allowing random access to the objects within a PDF file.

Protection of data in a PDF file consists of two steps: computation of a key to be used to encrypt data, and encryption of the data. The key is simply a string of bytes. Key length is depends on version of encryption algorithm used.

Version 1 of encryption algorithm was introduced in PDF 1.1 specification, and works with keys of exactly 40 bits in length (to satisfy old U.S. cryptographic export requirements). Version 2 and 3 of encryption algorithm defined in PDF 1.4 specification, and allows variable-length keys up to 64 and 128 bits in length respectively. Version 3 of algorithm still officially unpublished as an export requirement of the U.S. Department of Commerce.

All strings and streams within one object are encrypted with object-specific key. That key is calculated by passing document's key along with object ID and generation ID to the MD5 hash function. Several first bytes from MD5 output are treated as object key. Actual number of bytes used is equal to document key length increased by 5, but not more that 16 bytes. In Version 3 of algorithm bytes from object ID and generation ID are slightly transformed, and additional 4-bytes "sAJT" string passed to MD5 input.

When PDF viewer (like Adobe Acrobat Reader) displays encrypted document, the only things that viewer needs to know is document encryption key and encryption algorithm version. Version number is stored in Encryption dictionary, while document key should be provided to PDF viewer by security handler, which was used to encrypt the document. There are many different security handlers already exists, and anyone could develop new one (e.g. by using Acrobat SDK).

Now lets take a look at several widely used security handlers. Most of them are implemented as plug-ins for Adobe Acrobat Reader.

### **Standard security handler**

The most popular security handler is "Standard" one. Support for that handler is embedded in Adobe Acrobat and almost any PDF viewer.

PDF's Standard security handler allows two passwords to be specified for a document: an Owner password and a User password. Correctly supplying either password allows a user to open the document, decrypt it, and display it on the screen. Correctly supplying the Owner password will give full access to a document. Correctly supplying the User password will potentially provide a reduced set of operations that can be performed on the document. Access information in the document's Encryption dictionary specifies which of these operations, if any, may be performed. The Owner password must be supplied in order to change these restrictions or the passwords themselves.

The following operations are optionally permitted for Standard security handler 2 when a correct user password is provided:

- Modifying the document's contents
- Copying text and graphics from the document
- Adding or modifying text annotations and interactive form fields

000048

- Printing the document

The following additional operations are optionally permitted for Standard security handler 3 (announced in Acrobat 5) when a correct user password is provided:

- Form fill-in and sign document
- Text inspection for accessibility
- Document assembly, including insertion, rotation, and deletion of pages and creation of bookmarks and thumbnails
- Allow only printing that does not allow perfect digital copies, but which may also result in degradation of output quality

It is obvious that PDF cannot enforce the document access privileges specified in the encryption dictionary. So, if user password is empty or known, it is possible to calculate document encryption key and decrypt the document. That is exactly what Advanced PDF Password Recovery application developed by ElcomSoft does.

If neither user nor owner password is known, dictionary or brute-force attack could be performed. The following table displays testing speed for User and Owner passwords on documents protected with Standard security handler 2 (Acrobat 2, 3, 4, 5) and 3 (Acrobat 5 only). First line in each cell represents number of passwords that could be tested in one second on computer equipped with 450MHz Pentium III CPU. Second line reflects number of blocks to be passed through MD5 hash function and number of RC4 initializations required to test one password. All results were obtained with Advanced PDF Password Recovery Pro developed by ElcomSoft.

Handler type \ password type	User	Owner
Standard security handler 2	190,000 1×MD5 + 1×RC4	100,000 2×MD5 + 2×RC4
Standard security handler 3	3,250 51×MD5 + 20×RC4	1,610 102×MD5 + 40×RC4

Attack on password can not guarantee that document key will be found. But if document encrypted with 40-bits key, it is possible to find correct one by enumerating all possible keys. Single key testing (for Standard security handler 2) requires only one RC4 computation (initialization plus decryption of about 1 byte). Complete keys enumeration takes about 40 days on single 450MHz CPU.

Probably 40 days per document seems too long. So, there are several ways to reduce time needed to find document encryption key. First approach is based on parallel computing. For example, if 4 computers are available, document key will be found in 10 days.

When more than one document with Standard Security handler needs to be cracked, keys for all that documents could be found in one pass. That will require almost equal time for any number of documents because the most time-consuming operation (RC4 initializations) is performed once for each key to be tested, regardless of number of documents.

Finally, if large data storage is available, it is possible to create array of pre-computed data, which could be used to skip testing of some wrong keys. Every  $2^{16}$  bits (128 gigabytes) allows decrease number of keys to be tested. On the other hand, it is not possible to store all possible keys.

6700000

computers with 128 gigabytes of hard drive space each, will find the key in 64 times faster than ordinary single-CPU keys enumeration. Certainly, filling data storage with pre-computed data requires at least the same amount of time as key search for one document, but after that key for any document could be found in less than one day.

By the way, enumeration of all 40-bits keys could be performed not only for Standard security handler, but also for any security handler, which uses Version 1 of encryption algorithm. Such attack is sort of plain-text attack. Assuming that document creation date is stored as ASCII string, we could check appropriateness of document key by calculating object key and decrypting string with date. If resulted string looks like valid date string, than correct key is found. Each key testing requires one MD5 and one RC4 calculation. So, complete enumeration will take about two months on single 450MHz CPU. That time could be reduced only by parallel calculations. Multiple documents processing or pre-computed data is useless because object key is dependent on object number, and it is improbable that many documents will have creation date stored in object with the same number.

### *Rot13 handler*

Story of the Rot13 handler looks just ridiculous. That handler is used to protect reports created by New Paradigm Resources Group, Inc. (<http://www.nprg.com>). Each copy of their report costs about US\$3000, and documents protection looks adequate. Customer subscription includes hardware dongle, which needed to open any report. Every document is protected with password. But there is one odd thing...

Software Development Kit for Adobe Acrobat 4 contains sample code for two security plug-ins. And, as you probably already guessed, one of them is called "Rot13". That sample designed with several security flaws. The most significant one is that password is just checked, but not used in encryption, and all documents, protected with that handler, uses the same fixed document encryption key. That key is stored as string and could be found by text viewer in the body of the compiled plug-in. Definitely, plug-in used by New Paradigm Resources Group contains additional code to check dongle presence, and uses different encryption key. But key is still fixed and easily findable by text viewer. So, all that hacker needs to get full access to any report is modifying key constant, removing password checking from sample sources, and compiling the plug-in.

### *FileOpen handler*

FileOpen security handler was developed by FileOpen Systems (<http://www.fileopen.com>) and used in their software product called FileOpen Publisher™. FileOpen Publisher is unsurpassed in providing publishers with the most extensive and flexible access controls. The software includes tools for managing electronic subscriptions, authentication of concurrent users, document expiration, and locking to specific media (only from the web, only from CD-ROM, etc). FileOpen is also a pioneer in enabling printing restrictions on documents, so publishers can limit the number of printouts a user can make, or the period of time in which a document may be printed. A license to FileOpen Publisher costs US\$2500. Together, FileOpen and Adobe Acrobat 5.0 provide a complete, secure e-publishing solution.

That is advertising materials say. But how secure FileOpened documents in reality? FileOpen Publisher 2.3 and earlier encrypts all documents with the same constant key. In December 2000, ElcomSoft has published press release, which tells about flaws in FileOpen security handler. After that our technical director Vladimir Katalov has several conversations with President of FileOpen Systems Inc.

000050

That man started from complains on his cruel destiny and then asks to remove all mention of FileOpen vulnerability from ElcomSoft's web site, cease all discussion of FileOpen on all public newsgroups, and produce new version of Advanced PDF Password Recovery software with no functionality to decrypt FileOpen'ed documents. Certainly, he got the answer that it is impossible to fulfill his requests, but ElcomSoft could help FileOpen in making their plug-in more secure. On that FileOpen's representative said that he would take it into court if his requests would not be fulfilled. Probably his lawyer said that there are almost no chances, but since that time we did not hear any other threat from him.

Anyway, in January 2001 FileOpen has released version 2.4 of their Publisher software, which differs from previous version only in improved security. New version encrypts documents with variant keys, but document itself contains all necessary information to instantly calculate the key, and could be decrypted by new version of Advanced PDF Password Recovery. Probably they just grudge the moneys to hire security specialist and develop real protection.

### ***SoftLock handler***

SoftLock handler was developed by SoftLock Services, Inc. (<http://www.softlock.com>). SoftLock is an Adobe Acrobat plug-in that allows you to add security features to PDF documents. When the User opens a SoftLocked PDF, SoftLock calculates a unique number called a SoftLockID, which is based by default upon the User's Hard Drive. SoftLock then looks for the unique password, which is appropriate to that SoftLockID and PDF. If the correct password is present, PDF is unlocked; if it is not, customer is invited to purchase the correct password.

The approach used by SoftLock handler is not bad. But implementation corrupts the idea. Unlocking password length is exactly 8 characters. Any character than converted to one hexadecimal digit. Hence resulted password length is 32 bits. Moreover, 8 bits are used for password integrity check. So, effective password length is only 24 bits. It is easy to write a program that will find correct password by simple enumeration. All possible passwords could be tested by internal SoftLock's checking routine in about 30 hours on 450MHz CPU. Password checking algorithm is based on modified MD5 hash function. So, optimized version probably will find the password in minutes.

In May 1, 2001 SoftLock has claimed about a significant reduction of operations to avoid bankruptcy. Who knows, probably their troubles is related with careless in security plug-in development...

### ***Adobe Web Buy handler (PDF Merchant)***

In Acrobat 4.05 Adobe has implemented support for new security handler - Web Buy. That is the first handler, which allows key length greater than 40 bits. Web Buy handler required to view documents protected with PDF Merchant Digital Rights Management (DRM) technology. Customers are restricted from viewing document until they purchase a license. In most cases licensed document could be opened only from computer, it was bought for. To implement such functionality Web Buy handler sends to licensing server one or more identifiers like computer (processor) ID, storage volume numbers, user ID, etc. Server generates .RMF file (probably abbreviated Rights Management Format) and sends it to user. RMF file is XML document, which contains obfuscated PDF encryption key, document access permissions (e.g. Print) and certificate to check license validity.

There are three different Public RSA keys involved in document key calculation procedure. At first step 1024-bits Publisher's key, which stored in license certificate, is used to check license

validity. After that another 1024-bits key is used to check certificate validity. It looks like that key is owned by Adobe.

When license and certificate integrity is confirmed, Web Buy checks if document licensed for current computer configuration. There are one or more records in RMF file, which describes components to be tested. Each record contain test condition (equal, not equal, greater, etc.), Component Name (CPU, USERID, UTC, etc.), required Component ID, and encrypted key value. Records could be combined with logical operators "AND" and "OR". If this checking successfully passed, key value from some key is decrypted with the same Publisher's key that was used for license integrity checking, and decrypted one more time with 912 bits key, which seems to be owned by Adobe too.

Resulting 20-bytes value is combined with data from Encryption Dictionary and with Component ID (from validation record) by means of MD5 hash and XOR operation to get document encryption key.

Two RSA keys owned by Adobe (one for certificate generation and another for document key encryption) guarantees that publishers will pay Adobe for each document to be protected. And it seems impossible to falsify license by updating document permissions. But if anyone have access to pair of PDF and RMF file, it is possible to calculate document encryption key and decrypt the document, even without physical access to computer, the document was licensed for.

#### ***Adobe's Acrobat eBook Reader EBX handler (formerly GlassBook)***

Acrobat Reader is not only one PDF viewer with security support in the world. There was at least one more viewer - GlassBook Reader. But GlassBook Inc., development company of the GlassBook Reader, was acquired by Adobe, and viewer was renamed to Adobe's Acrobat eBook Reader. eBook Reader implements Electronic Book Exchange (EBX) protocol developed by EBX Workgroup (<http://www.ebxwg.org>) to manage document keys. Protocol specification is freely available, and describes data processing in details.

During activation of the Reader, RSA key pair is generated, and Public key is registered on server, while Private key retained by Reader. After buying license to some book Reader receives Voucher, which contain document key encrypted with Reader Public key, information about document permissions, and additional data which allows Voucher authenticity checking.

So, knowing Private key is necessary and enough to calculate document key and decrypt the document. In eBook Reader Private key stored encrypted by RC5 algorithm in Vouchers subdirectory. There are two ways to get RC5 key and therefore the Private key. RC5 key could be calculated as SHA1 hash of several computer-specific IDs (like CPUID and HDD volume ID). Another copy of RC5 key stored encrypted with another key, which is hard-coded in the body of Reader executable.

It is impossible to modify document permissions and falsify Voucher authenticity without knowing document key. But after extracting Private key and calculating document key it is very easy to generate Voucher with any permission and for any computer.

#### ***Arbitrary handler (obtaining encryption key from PDF viewer)***

Adobe Acrobat and Acrobat Reader of any version prior to 5.0 has no reverse-engineering protection at all. InterTrust DocBox plug-in supplied with Acrobat 5.0 uses simplest debugger-detection tricks and terminates Acrobat if debugger is present. Main executable file of the eBook Reader is encrypted and contains anti-debugging code, but both levels of protection could be

250000

remover by average-skilled hacker. Besides, none of those program checks for code integrity. So, it is very easy to do virtually anything with program by modifying its code in memory.

As was noted earlier, document key should pass through MD5 hash function every time when encryption key for some object calculated. So, all you need to get encryption key for any document, which could be opened in PDF viewer, is intercept MD5\_Update function and wait until key will be passed to it. There are several ideas that could help to find address of MD5\_Update function:

- MD5\_Update function resides not far from MD5\_Init function, which uses constants 0x67452301, 0xEFCDAB89, 0x98BADCFE and 0x10325476.
- MD5\_Update often called just after call to MD5\_Init function.
- MD5\_Update or some function called from MD5\_Update uses 64 constants defined in MD5 specification.

It is looks odd that in most cases Advanced eBook Processor it connected with Adobe eBook Reader only, because AEBPR has ability to get document encryption key and perform decryption for any PDF document which could be opened and viewed in Acrobat or Acrobat Reader.

### *Security flaw in Acrobat plug-ins certification*

There is another possible security flaw in products from Acrobat family. Adobe provides plug-ins SDK and plug-ins certification mechanism. Acrobat Reader loads only certified plug-ins. Each developer should sign Reader Integration Key License Agreement and pay fee of \$100 to obtain Reader Certification digital certificate from Adobe. This certificate could be used to Reader Certify any plug-in.

Also in some cases (for example, when opening documents protected with Adobe.WebBuy or InterTrust.DocBox security handler) Acrobat reloads itself to make sure that only plug-ins certified by Adobe are loaded. But certificate checking algorithm makes decision about certificate validity upon plug-in's Portable Executable header only. So, any correction in plug-in code will pass unnoticed. Moreover, it is possible to modify Adobe certified plug-in to load any other plug-in, and pass control to it. Hence, any plug-in could be loaded as if it was certified by Adobe, making certification completely useless.

### *Conclusion*

As you see, the most recent PDF specification provides encryption ability with key length up to 128 bits. Key generation algorithm could be implemented in any desirable way by means of security plug-in. But all that capabilities becomes useless if document could be opened for viewing. Moreover, several security plug-ins so poorly implemented that allows get the document key with little efforts.

The main problem of PDF encryption is that encryption key is passed to viewer, and document should be decrypted before displaying. So, there is some place where key is available, and another place where content is already decrypted. Implementing anti-reversing tricks or complicated cryptography will make key extraction harder, but not impossible.