Problem Set 7 Solutions

This problem set is due on *Tuesday, November 20, 2001* at the beginning of class. Note the abnormal due date. Late homeworks will *not* be accepted. You are to work on this problem set in groups of three or four people.

Mark the top of each sheet with your names, 6.857, the problem set number and question, and the date. **Type up your solutions** and be clear. Each problem should begin on a new sheet of paper. That is, you will turn in each problem on a separate pile of paper. **Cite** your sources of information.

Problem 7-1. Virus!

You have been hired by MegaScan, the leading provider of virus scanners, to write a special purpose subroutine to detect occurrences of the "UoyEvolI" virus, a particularly nasty polymorphic virus. Because of the way this virus inserts itself into the infected program, it isn't detected by the usual procedures (it doesn't execute when the infected program starts, but only some time later). Your subroutine will be run on all files to detect the UoyEvolI virus.

The UoyEvolI virus uses encryption to achieve polymorphism; the body of the virus is encrypted by exclusive-oring with a byte sequence S derived from an eight-byte secret key K that changes from instance to instance in a random way. The sequence S is derived by merely repeating over and over the given key K.

How would you approach this problem? What experiments would you run?

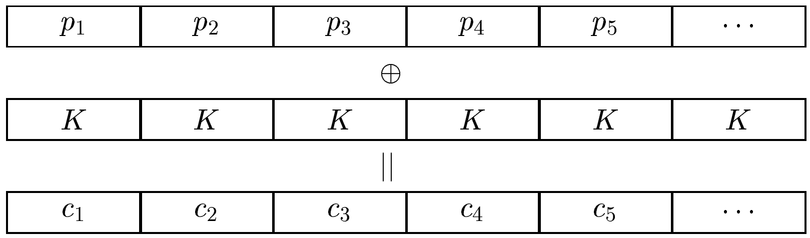
Solution: There were two types of neat solutions to this problem. Timo Burkard, John Gu, Kenny Yu and Massimo Mazza presented a solution in which the unencrypted virus code is xored with the suspected program at different offsets. If the resulting bytes have sections in which 8-byte chunks are repeated several times, then the program is probably infected.

The clever solution below, which uses a different technique, is presented by Petros Boufounos, Luciano Castagnola, and Nikos Michalakis.

The traditional virus scanning approach involves detecting a virus signature in the file it has infected. However, this approach will not work with our problem, since the code is scrambled. Still, we will show that we can form a kind of a signature for the virus, by exploiting the scrambling scheme.

The first step to detecting the virus is to figure out its operations and find its code. Since we have some form of the virus—e.g. if it infects executable code, then we have an infected executable file—we can run it in a well configured computer, equipped with all the appropriate debuggers and code analyzers, and properly isolated from the rest of the world. Indeed, we need to simulate a step by step execution of the virus within the debugger, at least up to the point where the code is unscrambled. That version of the code will serve as our starting point.

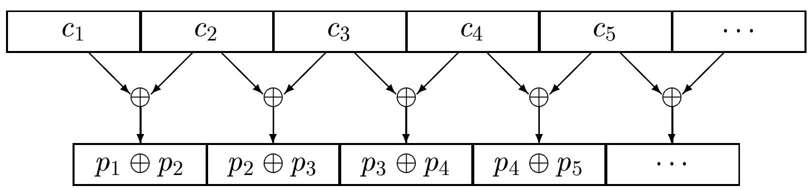
We will break the plaintext of the code in eight-byte blocks, denoted p_i, where i is the block number. We will also denote the corresponding scrambled blocks with $s_i = p_i \oplus K$. The encryption of the code is shown in the figure below:



It is obvious that $c_i = p_i \oplus K \Rightarrow c_i \oplus c_j = p_i \oplus p_j$, independent of K. Therefore we can create the block sequence:

$$(c_1 \oplus c_2, c_2 \oplus c_3, c_3 \oplus c_4, \ldots) = (p_1 \oplus p_2, p_2 \oplus p_3, p_3 \oplus p_4, \ldots),$$

as shown in the figure below:



This first N blocks of this sequence will be the signature of the virus, and we will use it for the detection. To scan if the file is infected we need to use a rolling XOR operation of the i^{th} byte with the $(i+8)^{th}$, i.e. XOR the file with itself left shifted by 8 bytes. Then we just need to check if the signature exists in the resulting bytestring (the efficient implementation would probably use an on-the-fly comparison and detection algorithm, operating straight on the rolling XOR operation; rather than calculating the XOR of the whole file with itself in one step and then scanning the output). We need to pick an N large enough to eliminate the false alarms on existing "clean" software (which we can do by testing the algorithm on installed libraries of known software).

Note that there are ways to improve the efficiency of this algorithm, without compromising accuracy (such as detecting repetitive patterns in the signature, or, if we know that the virus attaches at the beginning or the end of a file, only scan that part). Still, the main idea is the same, and these are exercises beyond the scope of this problem.

Furthermore, we should note that it is not necessary to get the unscrambled version of the code to obtain the signature. If we know which part of the file is the infected part, we can just perform the XOR operation straight on that part and try to determine a signature from there. The infected part of the file can be determined by comparing an infected file, with the corresponding uninfected one.

Problem 7-2. The DMCA Section 1201: Paracopyright

Niels Ferguson refused to release his break of HDCP[1] because of the Digital Millennium Copyright Act (DMCA). The Electronic Frontier Foundation (EFF) argues that abuse of the DMCA is forcing Niels to withhold research. If he publishes his break, he could be arrested when visiting the U.S. On the other hand, the American Association of Publishers (AAP) says that Niels is over-reacting.

Your job is to write two one-page essays. In the first essay, you are the senior intellectual property attorney for the EFF. Argue why Niels is not over-reacting to the DMCA. In the second essay, you are the vice president of legal and governmental affairs for the AAP. You are in favor of the DMCA. Argue why Niels is over-reacting.

Both the AAP and EFF representatives agree that in order to comply with the WIPO treaty, the U.S. government *must* implement adequate laws to protect against circumvention of copy protection technologies. Both representatives agree that the DMCA *had* good intentions. Beyond that, the representatives begin to disagree....

Each essay should give two clear points supporting the thesis. Essays should include an introductory paragraph, a concluding paragraph, and paragraphs to support your points. Each essay must fit on one page.

You'll probably have a harder time arguing the point of the AAP. However, there are legitimate points to both sides. You cannot simply disagree with the assignment.

Solution:

The essays were great. Many students (we think rightfully) pointed out that one side of the debate was flawed. The intent of this problem was to help you understand both sides of the DMCA argument. After all, to defeat an argument it's helpful to know what the opposition will say.

A discussion between the EFF and AAP actually happened on November 6, 2001 at the ACM Computer Communications and Security conference. The senior IP attorney for the EFF argued with the VP of legal and governmental affairs for the AAP. In the real debate, the parties essentially disagreed over fundamentals of how to fix abuse of the law. Both agreed that the DMCA has good intentions. Beyond that, the sides disagree. The AAP felt that court cases would eventually stem the abuse of the DMCA. The EFF argued that researchers essentially have too much as risk and would prefer not to become guinea pigs.

Petros Boufounos, Luciano Castagnola, and Nikos Michalakis wrote the following essays.

Essay 1 (EFF):

DMCA has serious implications to the right of free speech for professional cryptographers. Legal experts argue that scientific papers like the one Mr. Niels Ferguson wants to publish might be considered circumvention technology as DMCA defines it. Furthermore, the DMCA does not define who is responsible for protecting the researcher from getting sued for his research, so even if a company decides that it doesn't wish to prosecute the cryptographer for breaking their system, the government might and vice versa.

The DMCA contains a badly defined section, the anti-circumvention provision, that is the source of all controversy[2]. That section makes it illegal to break encryption that prevents someone from getting access to copyrighted electronic content, even if the break is not used for the purpose

[1] `http://www.macfergus.com/niels/dmca/cia.html`
[2] `http://www.anti-dmca.org/faq_local.html`

of illegally distributing the content. This section is not defined well enough to ensure that a cryptographer that breaks the encryption for research interests is not a criminal. In contrast, this actually facilitates criminals, since the designer of the secure system will never know of its weaknesses, unless he can receive feedback. Mr. Ferguson is right in pointing this out and we share his discontent.

To support our position even further, we bring forward the following argument. Intel corporation has decided not to prosecute Mr. Ferguson for breaking the HDCP. Nevertheless, that does not guarantee that the government will not prosecute him so he has every right to feel insecure on US ground[3]. But that's not where the problem ends. Suppose after he published his paper a record company decides that its interests were undermined by his break and decides to sue him for losing money[4]. Even if Mr. Ferguson wins eventually, he will still have gone through a serious distraction from his work, criminal accusations, and large legal bills. Therefore, he has every right to feel that at any point after his publication he could turn into a US enemy and in the best case never be able to travel to the US.

We conclude that our arguments show clearly why Mr. Ferguson is not over-reacting when he says that his freedom of speech is violated. The DMCA should not prevent cryptographers from freely publishing their ideas. Furthermore, they should not have to worry that their work will be considered illegal when their intentions are for the improvement of cryptography. Despite the good intentions of the lawmakers, in such cases, the DMCA constitutes a limit to free speech. The crime should be in the use of circumvention devices to illegally distribute content, and not on the creation of such devices for research or fair use purposes.

Essay 2 (AAP):

DMCA was created to protect the copyrights of digital media publishers. This is an essential extension of the copyright laws in the United States and a requirement for every country that supports the WIPO treaty. We believe that Mr. Niels Ferguson is over-reacting to the DMCA on his publication "Censorship in Action: why I don't publish my HDCP results". The DMCA was not created to limit the freedom of speech, nor suppress academic research.

Patricia Schroeder, CEO of AAP stated in her speech[5] on September 20th the following: "No reasonable person on the planet believes that a totally secure encryption system can be developed. Any encryption can be cracked, just as with patience and determination, any lock can be picked. The law permits an exemption for research, so that if you discover a weakness in my encryption system and inform me, that's legal." This is the belief of AAP and it is clear that Mr. Ferguson's intentions of publishing his paper are for the improvement of HDCP. Since, he has not created a circumvention system that can be used for cracking HDCP, he is not violating US law and therefore he should not be concerned with his safety while in the US.

As a second point to support our claims we present two contrasting examples that clearly show why Mr. Ferguson should not be concerned with the violation of the DMCA. The Russian hacker, Dmitri Sklyarov, was selling software on his web-site that facilitated the crack of the Adobe e-book Pro system. Anyone that visited his web-site was able to use his software to redistribute copyrighted

[3]In the well-known case of Dmitry Sklyarov, Adobe decided not to pursue the case further, but the government decided to do so.

[4]`http://lists.anti-dmca.org/pipermail/dmca_discuss/2001-August/000079.html`

[5]`http://www.publishers.org/press/092001speech.htm`

material to millions of others in the Internet. That *is* US law violation and prosecution was not directed against Dmitry's research, but against the creation and distribution of a circumvention system. On the other hand, Professor Felten of Princeton university is not sued for wanting to publish his work on breaking the SDMI water-mark system [6].

It is clear that Mr. Ferguson's case is more similar to the second rather than the first example and we believe that Mr. Ferguson has no reason to blame the DMCA for limiting freedom of speech.

Problem 7-3. It's time to rock around the clock

(a) Crack Egg

What is wrong with Louis' decision to implement the `f()` and `f'()` functions his way? Is this a rotten cipher?

Solution:

Shalini Agarwal, Steve Bull, Christine Karlovich, and Casey Muller turned in this solution. It was the fastest working code we found.

Louis' decision to implement the $f()$ and $f'()$ functions his way makes the Egg Shell (TM) smart card vulnerable to a timing attack. If an adversary is able to observe the running time of encryptions and decryptions, she may be able to extract information about the secret key bits based on the amount of time it takes to encrypt or decrypt various inputs.

In this implementation, one of our assumptions is that the inputs to the $f()$ function are independent. Suppose Eve observes $2N$ encryptions, and records the inputs and the time it took to do each encryption. Since the first part of the key is XOR'd with the plaintext before calling $f()$, we can determine the statistics of the running times based on whether or not the first bit of the input to $f()$ is 0 or 1 the first time it is called.

Suppose the first bit of at least half of the plaintext inputs Eve observes is 0 (similar argument for 1). Let X be the average time it took to encrypt each of N plaintexts whose first bit is 0. If the first bit of the key is 1, then the first bit of the first input to $f()$ will be 1. So
$\mathrm{E}[X] = 1 + \mathrm{E}[\text{time due to the rest of the bits in the first round}] + \mathrm{E}[\text{time due to last 2 rounds}] = 1 + 63 \cdot \frac{1}{2} + 32 \cdot 2 = 96\frac{1}{2}$.
Likewise, if the first bit of the key is 0, then
$\mathrm{E}[X] = 0 + \mathrm{E}[\text{time due to the rest of the bits in the first round}] + \mathrm{E}[\text{time due to last 2 rounds}] = 63 \cdot \frac{1}{2} + 32 \cdot 2 = 95\frac{1}{2}$.
In either case, $\mathrm{var}(X) = \frac{63 \cdot \frac{1}{4} + 16 \cdot 2}{N} = \frac{47\frac{3}{4}}{N}$ (this relies on independence of the encryption times). The Chebyshev Inequality tells us that $\Pr(|X - \mathrm{E}[X]| \geq c) \leq \frac{\mathrm{var}(X)}{c^2}$. We want the probability p that X differs from its expected value by more than 1/2 to be very small, $p \leq \frac{\mathrm{var}(X)}{(1/2)^2} = \frac{191}{N}$.

The probability that we make a correct decision on each bit in the key using a scheme like this is then $(1-p)$. So the probability we get the first 192 bits of the key correct is $(1-p)^{192}$. (If we get the first 192 bits correct, the last 64 bits come for free.) So we really want $(1-p)^{192} > P$, or $p < 1 - \sqrt[192]{P}$. So we can set $\frac{191}{N} = 1 - \sqrt[192]{P}$ to get the P we want, which gives $N = \frac{191}{1-\sqrt[192]{P}}$. For $P = 1 - (\frac{1}{10})^k$, $N \approx 3.67 \cdot 10^{k+4}$.

(b) Coding time!

[6] `http://www.newsbytes.com/news/01/166544.html`

Solution:

Shalini Agarwal, Steve Bull, Christine Karlovich, and Casey Muller donated this text along with their excellent code (available from the 6.857 Web page):

We implemented an attack that calls eggSmartCard.timeScramble() $3 \cdot 6272 + 1$ times. This is significantly less than what our theoretical result required to achieve any sort of confidence in our crack, but it seems to work well. The code is attached.