Using a Packet Sniffer to Analyze the Efficiency and Power of Encryption Techniques Used to Protect Data Over a Computer Network

Semyon Litvinov
Statistics Department/MCS Program
St. Cloud State University
slitvinov@stcloudstate.edu

Dennis Guster
Statistics Department/MCS Program
St. Cloud State University
guster@mcs.stcloudstate.edu

David Robinson
Statistics Department
St. Cloud State University
rob@stcloudstate.edu

Abdullah Alhamamah
Statistics Department/MCS Program
St. Cloud State University
alhamamah@hotmail.com

Abstract

The problem of defending against security attacks on internet traffic has become paramount in the last few years. One commonly used tool to combat hackers is encryption. Therefore, educating future networking students about the power of encryption and the overhead it places on network performance is important to their understanding of its place in the network design process. Therefore, this study examines the power of three types of encryption techniques and the additional overhead generated on a computer network when encrypted files are transmitted.

Introduction

The explosion of traffic on the Internet is the result of many users attempting to complete useful and private information exchanges. Unfortunately, just as the number of users has increased, so has the number of hackers attempting to compromise the total privacy most users seek. The accepted method of helping insure that privacy is some type of encryption. However, many people do not realize that making a decision to encrypt data can have an impact on network performance, and encryption techniques vary in power.

To clarify these relationships, this study will use a case study approach to investigate the effect encryption has on network throughput, randomness of payload, packet size, and packet inter-arrival times. A series of data blocks of various sizes will be identified and readied for transmission across a switched ethernet network. A monitor workstation (packet sniffer) will be placed on the network so that the characteristics of each packet set can be recorded and analyzed at a later time.

Each of the data blocks identified earlier will be transmitted twice. Once in normal unaltered form and once in encrypted form. Upon completion of the experiment overhead relating to both server and network infrastructure will be determined. This process will be repeated for each encryption algorithm studied.

The files used during the monitoring process will also be used to provide some insight into how powerful (hard to break) a given algorithm is. Encrypted payloads will be subjected to a test of randomness of characters. It is assumed, based on the theory of entropy, that data streams containing English text would have certain letters such as *e* or *a* appear more than *x* or *z* and, therefore, give potential code breakers insight into how to formulate their code-breaking strategies. Reducing or eliminating these patterns is critical to ensuring data privacy and, therefore, the test proposed will provide some insight into the power of algorithms utilized. Understanding these power differences is especially important for students just beginning their instructional career in networking.

Therefore, it is critical that these students develop an understanding of the interrelationships between encryption and computer network performance, specifically the additional overhead associated with encrypting files or data streams.

There is little doubt that computer security within the internet is one of the premiere problems in computer networking today [1-8]. Too often students view the problem rather simplistically, that is, they feel that an encryption program on the sending side matched with a decryption program on the other side is the prescription to cure the hacking problems. They give little thought to the development of a comprehensive security plan and are unaware of any adverse effects implementation of encryption may cause. Furthermore, they have little concept regarding the effectiveness of various encryption techniques.

Performance

In most cases, employing encryption has an adverse effect on network performance. The level of this degradation is a function of the encryption algorithm employed. For simple techniques such as character substitution there is little or no overhead. However, for today's more secure techniques such as RSA (Rivest, Shamir & Adleman) there is significant overhead. In fact, it can be expected that a file encrypted in RSA will be several times larger than the original unencrypted version. This increase in file size increases the volume of information that must be transmitted over the network. The degree to which network performance will be influenced is related to the volume of information transmitted and how the interarrival times of the packets carrying that information are distributed. Generally speaking, lightly loaded networks are going to be less affected than heavily loaded networks. Also at issue is how well the traffic matches the MTU (maximum transmission unit) of the network infrastructure. For example, ethernet has a limitation of approximately 1500 bytes. If the encrypted file does not segment well in this environment, it results in an increase of the number of packets sent beyond the expected ratio.

In turn, this situation plus the larger number of expected packets based on the encrypted file's larger size could easily generate a much smaller packet interarrival mean. Also, the distribution of this packet stream could provide further risk to the goal of processing information in a timely manner.

Power of Encryption

There are a wide variety of encryption algorithms of varying degrees of sophistication. No matter their degree of sophistication, a common goal is to make it too difficult and too time consuming for would-be hackers to break them. Historically, one tool employed was intuition based on the expected frequency of letters used in the language of question. this method draws on the theory of entrophy [9]. For example, in English, it would be expected that "e" would appear 12.7 percent and a little-used character such as "z" would appear .07 percent of the time. This knowledge can be a powerful decryption tool in simple substitution and cipher methods, but is less powerful in more sophisticated key-based methods such as RSA.

Exploring this concept is an excellent starting point of analysis for students just beginning the study of cryptography. It is important for them to understand the added sophistication algorithms such as RSA and DES provide beyond substitutions and ciphers.

One way to approach this beyond explaining the formulas is to calculate a coefficient of randomness within the original and encrypted versions of that file. Operating under the assumption that the original text will follow expected frequency of letters within the English language. To accomplish this goal, two approaches were used. First, a series of four files were compared to a uniform distribution. These files were as follows:

an encrypted text file contain 1402 (non-blank) characters
that same file encrypted by substitution also 1402 characters
that same file encrypted by cipher also containing 1402 characters
that same file encrypted by RSA containing 2125 characters

The chi square statistic employed to determine compliance to a uniform distribution revealed that none of the files followed a uniform distribution, that is, a distribution in which all characters appeared with equal frequency. The logic being that if the character distribution is truly uniform, then intuition through enthrophy cannot be employed to break the code. Table 1 depicts the chi square values obtained from the analysis. Although all were statistically significant, the magnitudes of the values reveal that certain encryption methods resulted in a file that was more uniform than others. Specifically, the RSA was closest to the uniform while the encrypted and substitution file both deviated equally from that distribution. While the cipher distribution fell somewhere between the substitution and the RSA. Therefore, one could conclude that if entrophy were used as the decoding method, that the RSA would be the most difficult to decode since its characters are the most uniformly distributed of the three encryption techniques.

Table 1
Compliance to a Uniform Distribution

	# of characters	chi square value	sig
unencrypted	1402	14975	.001
substitution	1402	14975	.001
cipher	1402	3757	.001
RSA	2125	1639	.001

Although comparisons to uniform distributions yield some useful evidence concerning the randomness of characters, the process lacks sophistication and it is difficult to make interval level comparison with the chi square statistic. Therefore, the following algorithm is offered to ascertain randomness within data files. This example is presented in binary for the sake of simplicity and is based on an idea posed by Michael Guysinsky currently at Tufts University.

Methodology for Coefficient of Encryption Power

Let

$$W = \overline{x_1 x_2 \ldots x_N} , \quad x_i \in \{0,1\}$$

be a binary word to be examined for “randomness”. If we introduce a notation

$$W_m^n = \overline{x_m x_{m+1} \ldots x_n} ,$$

with $m \le n \le N$, then W_m^n is a subword of W, and we have $W = W_1^N$.
Our inspection is based on the following procedure. With every $n = 3,...,N$ we associate a number U_n measuring how "expected" is x_n with respect to the preceding word W_1^{n-1}:

$$U_n = \sum_{m=2}^{n-1} \frac{\#\left[W_m^{n-1} / x_n \subset W_1^{n-1}\right]}{\#\left[W_m^{n-1} \subset W_1^{n-1}\right]},$$

where

$$\#\left[W_m^{n-1} \subset W_1^{n-1}\right] \text{ and } \#\left[W_m^{n-1} / x_n \subset W_1^{n-1}\right]$$

are, respectively, the number of subwords of the form U_m^{n-1} in W_1^{n-1} and the number of those words among them followed by x_n.

Now we sum the numbers U_n up which yields the formula

$$U_W = \sum_{n=3}^{N} \frac{1}{n} \sum_{m=2}^{n-1} \frac{\#\left[W_m^{n-1} / x_n \subset W_1^{n-1}\right]}{\#\left[W_m^{n-1} \subset W_1^{n-1}\right]}.$$

Note that the coefficient n^{-1} is a weight and is designed to balance impacts from longer subwords of W.

Hypothesis. *The* <u>*smaller*</u> U_W *is, the more random is the word* W.

Example. Let $W_1 = 10010010$, and $W_2 = 11010010$. In both cases, $N = 8$.

For W_1, we get

$$U_3 = U_4 = 0,\ U_5 = \frac{1}{2},\ U_6 = \frac{1}{3} + \frac{1}{2} = \frac{5}{6},\ U_7 = \frac{1}{4} + \frac{1}{2} + \frac{1}{2} = \frac{5}{4},\ U_8 = \frac{2}{3} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} = \frac{13}{6}$$

which implies that

$$U_{W_1} = \frac{1}{5} \cdot \frac{1}{2} + \frac{1}{6} \cdot \frac{5}{6} + \frac{1}{7} \cdot \frac{5}{4} + \frac{1}{8} \cdot \frac{13}{6} \approx .69$$

For W_2,

$U_3 = U_4 = U_6 = 0$, $U_5 = U_7 = \frac{1}{3}$, $U_8 = \frac{2}{4} + \frac{1}{2} = 1$, therefore

$$U_{W_2} = \left(\frac{1}{5} + \frac{1}{7}\right) \cdot \frac{1}{3} + \frac{1}{8} \cdot 1 \approx .24$$

As it was expected, $U_{W_2} < U_{W_1}$.

Table 2 presents the value of the coefficients obtained when applying the algorithm to the same files used in Table 1. For the most part, the magnitude is what would be expected with the exception of the substitution file. That is, the value 42.35 obtained would be expected to be lower than that of the unencrypted file. However, this is not totally unexpected in that although a method of encryption substitution lacks sophistication. In fact, when compared to a uniform distribution, the substitution method yielded the same chi square as the unencrypted file.

Table 2
Application of the Encryption Power Coefficient

unencrypted	37.15
substitution	42.35
cipher	36.88
RSA	5.99

However, the rest of the values appear to scale nicely as expected. In fact, the value of 5.99 provides further indication of its power beyond the other methods examined in Table 2. Although the results of the application of this algorithm are encouraging, it still needs to be tested again. These tests should include numerous files encrypted by algorithms of known sophistication before its true validity can be established.

The Effect of Encryption in the Packet Stream

A reexamination of Table 1 reveals that some encryption techniques, in this case RSA, result in a much larger file when encrypted. In the Table 1 example, the unencrypted file was 1402 while the encrypted file was 2125 bytes. In most cases users are more than willing to accept the trade off of additional bytes to transmit for better security.

However, what effect does this additional overhead pose upon network performance? To help answer this question, the experiment using a packet sniffer was undertaken. First and standard text file containing 566 KBs was selected, and a RSA encrypted version was compiled. The encrypted version contained 1.1 MBs. Files of this size were used instead of the ones from Table 1 so that multiple data packets would be required to transmit the data instead of the one or two packets required for the Table 1 files (assuming 1500 MTU for ethernet).

First, the unencrypted file was transmitted from a samba client to a samba server. During this transmission a packet sniffer connected to the same collision domain trapped and monitored all packet traffic between these nodes and associated processes. This traffic was logged for further analysis. The same procedure was also applied to the encrypted file.

Table 3 displays the results of this packet traffic. As would be expected, the session length for the file transfer was less for the unencrypted file, 8.8 seconds versus 10.4 seconds for the encrypted file. these values may be somewhat misleading since there was other packet traffic running at the time of both sessions. It is doubtful that this background noise affected both sessions equally.

Table 3
Packet Traffic Trends

Session Duration	8.8782_{sec}	10.4184_{sec}
# of overhead	133	441
#of data	96	414
# of overhead bytes	9466	26460
# of data bytes	139008	599472

The number of overhead packets for the encrypted files was more than three times greater than the unencrypted file while its original file size was only one-half the encrypted file. There is a similar disparity in the number of data packets as well. However, the number of overhead bytes comes close to the expected two to one ratio.

The number of data bytes poses an interesting question – that is, why in both cases are they smaller than the original file size? The answer lies in the manner in which the files are packetized by SMB client. In this method a line of text is loaded in the packet until a carriage return and a line feed are encountered. Upon detection of these characters the rest of the line is ignored, and the packetization process continues with the next line. This process of ignoring the rest of the line thereby compresses out blanks and reduces the number of bytes that need to be transmitted across the network. It is interesting to note that there were about four times as many bytes that had to be transmitted in the encrypted file which may be indicative in this case that the compression was less efficient.

Summary and Conclusions

It is clear that a little experimentation can verify what is expected theoretically and at the same time be useful to help to explain complex networking/encryption concepts to students. In this paper a quick analysis of file encryption techniques determined that cipher was more powerful than substitution and the RSA technique the best of three techniques examined. This basic process could be expanded to include a number of other and more powerful algorithms if desired.

More sophistication could also be added to the packet stream analysis. It is very difficult to get students to realize that there is a significant amount of overhead taking place in network transmission. In fact, in the example provided there were more overhead than data packets. Although as a percentage of bytes transmitted the overhead packets only made up 4 to 6 percent of the traffic, their presence is still worth noting in network design activities.

References Cited

1. Earls, Alan R. “Between the Cracks”. Computer World, February 1997: http://www.computerworld.com

2. Ernst & Young. Information Security Survey: Analysis of Trends, Issues & Practices: 1996.

3. O’Higgins, Brian. “Intranet Security: Beyond Firewalls”. Electronic Commerce World, March, 1997: 14.

4. Strassman, Paul. “What’s the Best IS Defense? Being Prepared”. Computer World. February, 1997.

5. Vacca, John. Internet Security Secrets. Foster City, CA: IDG Books Inc., 1996.

6. Wood, Charles C. “Policies from the Ground Up”. Info Security News. January, 1997.

7. Bhimani, Anish. “Securing the Commercial Internet”. Communications of the ACM, 39:6, 1996.

8. Fink, D. Information Technology Security – Managing Challenges and Creating Opportunities, CCH Publishers, Sydney, 1997.

9. Hankerson, D.R., et.al. Coding Theory and Cryptography: The Essentials. New York, Marcel Dekker, Inc., 2000.

10. Mel, H.X. & Baker, Doris. Cryptography Decrypted. Boston: Addison-Wesley, 2001.