Latency, Packet Loss and Encryption using DES with a VPN

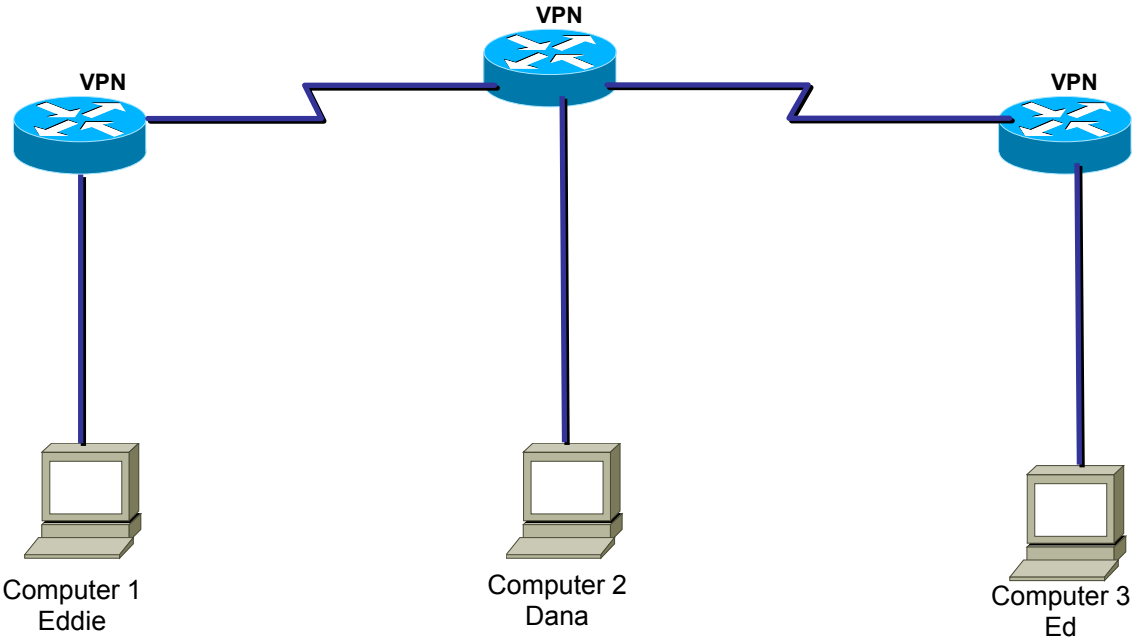
By
Eddie Sutton

Security is at the top of the list in today's world concerns. The need for encryption is extremely vital in the business place. We start talking about the need for encryption, 128, 192, 256 and higher bit encryption methods. Does this affect the transmission rates of file transfer enough to warrant the use of higher bit encryption on semi-secure documents? I am conducting an experiment to show what, if any, the latency is when dealing with the higher end encryption methods. I used a variable of no encryption to benchmark the time and speed then used DES encryption with the VPN and finally 3DES. This field experiment will prove if there is latency when dealing with the higher encryptions. I feel this test will prove there is latency with this.

The equipment being used in this test is as follows. I will be using three computers. Two of the computers are running Windows XP, the other is running Windows 2000 Pro. This was the luck of the draw. My computer, my dad's computer and Dana's, a friend of mine, were already all running this particular operating system. We all use broadband as an Internet connection. I use Road Runner while Ed and Dana use Earth Link. I am listing what is important in the computers dealing with the sending and receiving of files. My computer specs are as follows, a 1-gig processor, 512 MB of memory, and a 100 MB NIC. Ed's computer has a 1.8-gig processor, 512 MB of memory, and a 100 MB NIC. Dana's computer has a 2.4-gig processor, 1024 MB of memory, and a 100 MB NIC. Ed and I are using a LinkSys router while Dana is using a NetGear router. The LinkSys routers are four-port routers. Dana is using a NetGear

Model FVS114. There was a Belkin router in the mix which was not robust enough to handle what I wanted it to do so I bought the NetGear Model FVS114 router to replace it for my testing purposes. The specs can be found here http://www.netgear.com/products/details /FVS114.php. We will be setting up a VPN between the three of us to use as the testing ground of my experiment. A VPN is a private network that uses a public network (usually the Internet) to connect remote sites or users together. Instead of using a dedicated, real-world connection such as leased line, a VPN uses "virtual" connections routed through the Internet from the company's private network to the remote site or employee. (http://computer.howstuffworks.com/vpn.htm) The layout of the test computers and routers looks something like this.

VPN Computer Layout



Computers Specs Used in the Field Test

	Eddie Computer 1	Ed Computer 2	Dana Computer 3
Processor Speed	1-gig	1.8-gig	2.4-gig
RAM	512MB	512MB	1024MB
NIC	100MB	100MB	100MB
Internet Speed	Cable	Cable	Cable

Before I jump right in to the test and results lets discuss what encryption, DES and 3DES encryption is. Information security is provided on computers and over the Internet by a variety of methods. A simple but straightforward security method is to only keep sensitive information on removable storage media like floppy disks. But the most popular forms of security all rely on encryption, the process of encoding information in such a way that only the person (or computer) with the key can decode it. Computer encryption is based on the science of cryptography, which has been used throughout history. Before the digital age, the biggest users of cryptography were governments, particularly for military purposes. The existence of coded messages has been verified as far back as the Roman Empire. But most forms of cryptography in use these days rely on computers, simply because a human-based code is too easy for a computer to crack. Most computer encryption systems belong in one of two categories: Symmetric-key encryption and Public-key encryption. (http://computer.howstuffworks.com/encryption.htm)

In symmetric-key encryption, each computer has a secret key (code) that it can use to encrypt a packet of information before it is sent over the network to another

computer. Symmetric-key requires that you know which computers will be talking to each other so you can install the key on each one. Symmetric-key encryption is essentially the same as a secret code that each of the two computers must know in order to decode the information. The code provides the key to decoding the message. Think of it like this: You create a coded message to send to a friend in which each letter is substituted with the letter that is two down from it in the alphabet. So "A" becomes "C," and "B" becomes "D". You have already told a trusted friend that the code is "Shift by 2". Your friend gets the message and decodes it. Anyone else who sees the message will see only nonsense.
(http://computer.howstuffworks.com/encryption.htm)

Public-key encryption uses a combination of a private key and a public key. Only your computer knows the private key, while your computer to any computer that wants to communicate securely with it gives the public key. To decode an encrypted message, a computer must use the public key, provided by the originating computer, and its own private key. A very popular public-key encryption utility is called Pretty Good Privacy (PGP), which allows you to encrypt almost anything. To implement public-key encryption on a large scale, such as a secure Web server might need, requires a different approach. This is where digital certificates come in. A digital certificate is basically a bit of information that says that the Web server is trusted by an independent source known as a certificate authority. The certificate authority acts as a middleman that both computers trust. It confirms that each computer is in fact who it says it is, and then provides the public keys of each computer to the other.
(http://computer.howstuffworks.com/encryption.htm)

The key in public-key encryption is based on a hash value. This is a value that is computed from a base input number using a hashing algorithm. Essentially, the hash value is a summary of the original value. The important thing about a hash value is that it is nearly impossible to derive the original input number without knowing the data used to create the hash value. You can see how hard it would be to determine that the value 1,525,381 came from the multiplication of 10,667 and 143. But if you knew that the multiplier was 143, then it would be very easy to calculate the value 10,667. Public-key encryption is actually much more complex than this example, but that is the basic idea. Public keys generally use complex algorithms and very large hash values for encrypting, including 40-bit or even 128-bit numbers. A 128-bit number has a possible 2^{128} or 3,402,823,669,209,384,634,633,746,074,300,000,000,000,000,000,000,000,000,000,000,000,000,000 different combinations! This would be like trying to find one particular grain of sand in the Sahara Desert. (http://computer.howstuffworks.com/encryption5.htm) That is a crash course on encryption or at least what we need to know for this paper.

DES encryption or the Data Encryption Standard was jointly developed in 1974 by IBM and the U.S. government (US patent 3,962,539) to set a standard that everyone could use to securely communicate with each other. It operates on blocks of 64 bits using a secret key that is 56 bits long. The original proposal used a secret key that was 64 bits long. It is widely believed that the removal of these 8 bits from the key was done to make it possible for U.S. government agencies to secretly crack messages. DES started out as the "Lucifer" algorithm developed by IBM. The US National Agency (NSA) made several modifications, after which it was adopted as Federal Information Processing Standard (FIPS) standard 46-3 and ANSI standard X3.92. (http://www.iusmentis.com/

technology/encryption/des/) This method of encryption has been cracked with in 24 hours so it is not considered the best available method of encryption.

DES encryption works by taking blocks of the message and encrypting it in 16 rounds or steps. Encryption of a block of the message takes place in 16 stages or rounds. From the input key, sixteen 48 bit keys are generated, one for each round. In each round, eight so-called S-boxes are used. These S-boxes are fixed in the specification of the standard. Using the S-boxes, groups of six bits are mapped to groups of four bits. The contents of these S-boxes have been determined by the U.S. National Security Agency (NSA). The block of the message is divided into two halves. The right half is expanded from 32 to 48 bits using another fixed table. The result is combined with the sub key for that round using the XOR operation. Using the S-boxes the 48 resulting bits are then transformed again to 32 bits, which are subsequently permutated again using yet another fixed table. This by now thoroughly shuffled right half is now combined with the left half using the XOR operation. (http://www.iusmentis.com/technology/encryption/des/) DES is vulnerable due to only having a 56 bit key size. We will discuss the higher end DES encryption method 3DES.

Triple DES is a block cipher formed from the Data Encryption Standard (DES) cipher. It was developed by Walter Tuchman (the leader of the DES development team at IBM) and is specified in FIPS Pub 46-3. There are several ways to use DES three times; not all are Triple-DES and not all are as secure. Triple-DES is defined as performing a DES encryption, then a DES decryption, and then a DES encryption again. The formula used to represent this is $C = DES_{k3}(DES^{-1}_{k2}(DES_{k1}(P)))$. (http://en.wikipedia.org/wiki/

Triple_DES) Triple-DES has a key length of 168-bits (three 56-bit DES keys), but because of the meet-in-the middle attack it has an effective key size of 112 bits. With the 24 parity bits (8 parity bits per DES key), Triple-DES has a total storage length of 192 bits. A variant, called two-key triple-DES, uses $k_1 = k_3$, thus reducing the key size to 112 bits and the storage length to 128 bits. However, this mode is susceptible to certain chosen-plaintext or known-plaintext attacks. These attacks are highly impractical, but they are a weakness of this variant. (http://en.wikipedia.org/wiki/Triple_DES) As stated by the Forums.Databasejournal "Triple DES or DES-3 is an enhancement over the existing DES standard. DES-3 encrypts each block three times with the DES. The algorithm uses either two or three different 56-bit algorithm, using either two or three different 56-bit keys. This approach yields effective key lengths of 112 or 168 bits".

Because Triple-DES applies the DES algorithm three times (hence the name), Triple-DES takes three times as long as standard DES. Decryption using Triple-DES is the same as the encryption, except it is executed in reverse. (http://www.iusmentis .com/technology/encryption/des/) Triple DES was the answer to many of the shortcomings of DES. Since it is based on the DES algorithm, it is very easy to modify existing software to use Triple DES. It also has the advantage of proven reliability and a longer key length that eliminates many of the shortcut attacks that can be used to reduce the amount of time it takes to break DES. However, even this more powerful version of DES may not be strong enough to protect data for very much longer. The DES algorithm itself has become obsolete and is in need of replacement. To this end the National Institute of Standards and Technology (NIST) is holding a competition to develop the Advanced Encryption Standard (AES) as a replacement for DES. AES probably will

surpass 3DES in the near future but I wanted to test DES and 3DES encryption standard to really see what it does in action and why it may be replaced with AES shortly.

I checked the routers out to make sure they were 3DES encryption compatible. It worked out that two of the three were so I went out and bought a replacement Net Gear router that is actually better functionality wise than the two LinkSys routers already in play. The NetGear router is quite robust. I was very surprised with this fact. I had not used NetGear products in a few years and the company seems to have caught up with the market. We put the NetGeat router at Dana's to be the central VPN point.

I am using a 5 MB file to use as the testing file. I felt a 5 MB file would be ample for me to clock the speed and time in the transfer. I will, as stated earlier, start the test using no encryption to benchmark the speed and time it takes to transfer the file across the VPN. Then I will encrypt the VPN with DES and send the file, likewise with the 3DES encryption. I will start the test using Dana's computer as the sending and mine as the receiving computer, then send it back. I will send and receive the file to and from all three computers as the sender and receiver. The DES encryption was 786 bit and the 3DES was at 1096 encryption. I am very interested to see if there is any difference in transfer rates with the higher encryption, 3DES, versus the DES and with no encryption. I feel the finds should show a marked increase in transfer rates for the DES and 3DES times. I am also interested in seeing how much bigger the file becomes by adding the different encryptions to it.

After playing around with the VPN to get it connected I finally finished the testing. I'm not sure if it was the different routers in the mix or not, I really wouldn't

think that would be it, I got the VPN up and running. With the tests completed I will show my results and discuss them.

Data Transfer Speeds and Times

Unencrypted 5MB

From	To	Time	Avg Ping Rate
Dana	Ed	118sec	17ms
Dana	Eddie	121sec	20ms
Eddie	Ed	120sec	19ms

Data Transfer Speeds and Times

DES 5MB

From	To	Time	Avg Ping Rate
Dana	Ed	165sec	18ms
Dana	Eddie	185sec	21ms
Eddie	Ed	175sec	22ms

Data Transfer Speeds and Times

3DES 5MB

From	To	Time	Avg Ping Rate
Dana	Ed	168sec	19ms
Dana	Eddie	189sec	22ms
Eddie	Ed	175sec	21ms

Well, I must say I am surprised at the findings. I really thought the higher level encryptions would significantly slow the transfer rates down compared to the unencrypted transfers and, while it did some what, it mainly slowed the time by a fifty percent increase from my fastest transfer rate, 120ms, to the slowest, 189. The ping rate was excellent. Encrypted or unencrypted, it seemed to not effect the ping. Now fifty percent is quite notable but using unencrypted transfers versus 3DES encrypted transfers I really expected the transfer rates to go up quite a bit higher. Now when dealing with a fifty MB size file this may get noticeable, but with the five MB file it was not noticeable.

The latency was prevalent but to no degree the latency I thought it would be. The DES rates closely match the 3DES rates which I thought the higher encryption, 3DES, would markedly make the file size larger and it did not. The test proves, to me at least, that the VPN speed with encryption is limited only by the computer's upload speed, which is the case in any network sending and receiving data. I also would like to mention there was no packet loss in these transfers, although I had not expected any.

Overall I was very impressed with the VPN performance dealing with the higher encryption levels. The encrypted VPN was very quick and responsive, as the ping results show. I see no reason for any business, or for that reason, individual that transmits important data not to use encryption if their reasoning deals with the slower rates of transfer and latency. As my findings show, there is not enough degradation in the VPN performance to warrant not using it with encryption at any level. After thinking about the transfer rates, which surprised me, I realized that with or without encryption the transfer rates should be close directly due to the Internet connection it works through. If I was on dial up it should reflect the unencrypted rates plus a bit more due to the fact of the encryption header. This test really shed a lot of light on the field of VPNs for me. I understood in theory why they were very important to the business world, but while testing and then seeing the results of my test I see the importance of the VPN in the workplace. I am now in the process of setting up a VPN for good with the people who were gracious enough to let me play with their computers. I feel the resource sharing between the three of us can benefit us greatly.

Sources and References

Anonymous author, Data Encryption facilities in Daffodil DB, from forums.databasejournals.com http://forums.databasejournal.com/archive/index.php/t-36403.html

Tyson, J (year unknown). How encryption works Retrieved on July 1, 2005 from How stuffworks.com http://computer.howstuffworks.com/encryption.htm

Staff writer for Tropsoft.com, DES Encryption. Retrieved June 25, 2005 from Tropsoft.com http://www.tropsoft.com/strongenc/des.htm

Staff writer for Tropsoft.com, 3DES Encryption. Retrieved June 25, 2005 from Tropsoft.com http://www.tropsoft.com/strongenc/des3.htm

Staff writer for Wikipedia.com, Data Encryption Standards. Retrieved on June 25, 2005 from Wikipedia.com http://en.wikipedia.org/wiki/DES

Anonymous author, Encryption Software Group, Retrieved on June 25, 2005 from des-rsa-encryption-software-cryptography-group.com http://www.des-rsa-encryption-software-cryptography-group.com/

Englefriet, A (2003), Ius Mentis, December 30, 2003, Retrieved June 25, 2005 from IusMentis.com http://www.iusmentis.com/technology/encryption/des/

Yu, J (1998), Cracking the DES encryption, Retrieved June 28, 2005 from ITAudit http://www.theiia.org/itaudit/index.cfm?fuseaction=forum&fid=93

Coppersmith, D, Johnson, D.B., Maytas, S.M. (1996) IBM Journal of Research Vol 40 Number 2, 1996, Retrieved June 28, 2005 from research.ibm.com/journal http://www.research.ibm.com/journal/rd/402/coppersmith.html

Sullivan, C., (2003) Xcell Journal Online Article, Retrieved June 29, 2005 from Xcell Journal Online http://www.xilinx.com/publications/xcellonline/xcell_45/xc_recon45.htm

Pasham V., Trimberger, S., (2001) Xilinx Retrieved June 28, 2005 from Xilinx Aug 03, 2001 High-Speed DES and Triple DES Encryptor/Decryptor http://www.xilinx.com/bvdocs/appnotes/xapp270.pdf

Alderson, B., (2001) NetAnalyst Training Retrieved June 29, 2005 from Finding latency sources in VPN and other Internet solutions February 2001, http://www.pmg.com/tip_archive/01_02.htm