Proceedings of the 4th Winona Computer Science Undergraduate Research Symposium

April , 2004

Table of Contents

Applying Steganography to Standard Network Traffic

Paul Miller
Saint Mary's University of Minnesota
700 Terrace Heights
Winona, MN 55987
pmmill00@smumn.edu

ABSTRACT
We examine a conventional but often overlooked application of steganography, establishing a covert communication channel. Due in part to the explosion of the digital watermarking discipline, most work in the field is done with non-volatile digital media files such as pictures, audio, movies, and software. However, such media are inefficient and insecure for the purpose of establishing a covert channel. We argue that a volatile medium should be chosen for this application, and compare two techniques, TCP header and HTTP request hiding methods. We found that the TCP header method provided far better security, while the HTTP request method provided a much larger bandwidth.

Categories and Subject Descriptors
D.2.11 [**Software Engineering**]: Software Architectures – *information hiding.*

General Terms
Algorithms, Performance, Design, Security.

Keywords
Steganography, Steganalysis, Covert Channels, Information Hiding.

1. INTRODUCTION
Historically, there have been two separate and distinct methods of protecting messages from unintended recipients: cryptography and steganography. The overall goal of cryptography is to make the message *unreadable*; the overall goal of steganography is to make the message *undetectable.* Due to the differing goals of the two methods, two distinct paradigms are used. While cryptography focuses on obscuring the content of the message, steganography focuses on hiding the existence of the message altogether. Cryptanalysis and steganalysis are terms referring to methods designed to analyze and break (understand/find the message) cryptography and steganography, respectively.

One of the first historical methods of protecting communications was a form of steganography; a message was hidden inside of a hare and delivered by a man dressed as a hunter. History delivers our first cryptographer as well, Julius Caesar, who was also attempting to protect a message. From that point on, cryptography became the preferred method for the transmission of messages; it has since been widely used and highly developed. Cryptology as we know it today probably has its beginnings in Arthur Scherbius' Enigma machine and the cryptanalysis methods and machines that broke its encryption. A valuable lesson can be learned from this history – merely encrypting a message is not always enough; sometimes it is also important to conceal the existence of the communication itself.

So how does steganography work? Since the practice centers on the hiding of messages in an overt medium, a critical step in the development of a steganographic technique is the selection of this medium. The overt medium is the container the message is hidden inside of; it is the face of the communication that the world can see. In our historical example, the hare is the overt medium. In the computing age, most work is done with images, audio files, and source code. The steganographic message (often a digital watermark) is often dispersed across the file and embedded into least significant bits, or hidden inside data which has little to no affect on the file. When done properly, the overt medium still functions in its original capacity, and there is no observable change in the file for steganalysts to detect. Many steganalysis methods rely heavily on statistical analysis of the file to determine if a hidden message is present [1, 2].

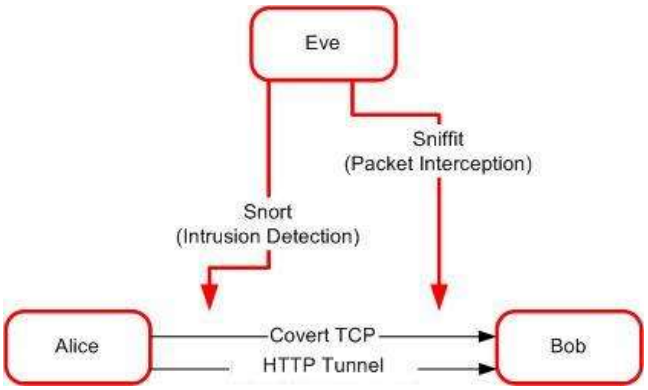


Figure 1. Our Research Model

Proceedings of the 4th Winona Computer Science Undergraduate Research Seminar, April 20–21, 2004, Winona, MN, US, Copyright 2004.

Let us consider the classic cryptography/steganography example of Alice, Bob, and Eve (Figure 1). The covert channel exists between Alice and Bob, and Eve is the attacker. First, using non-volatile media as an overt medium is often suspicious in and of itself [3]. Assuming the trading of pictures or sound files is discouraged or prohibited on this network as it is on many networks, the trading of JPEG"s or MP3"s alone would be cause for suspicion and investigation by Eve, something Alice and Bob do not want. Secondly, all of these media are non-volatile, meaning they must be stored permanently on a device (such as a hard drive), making it much easier for Eve to determine if a covert channel exists in these media and also to extract the message using steganalysis techniques. These files can be easily recovered; when comparing the two different versions (one with a hidden message and one without), proving the existence of the hidden message and extracting it becomes trivial. Finally, as new steganalysis techniques for these media have been introduced, detection rates have risen significantly [1, 2]. Thus, a new medium, less suspicious and more volatile, must be chosen for this application of steganography.

So what medium exists on all networks and is purely volatile? None; however, there are protocols such as TCP, ICMP, UDP, and HTTP (information on the specifics of these protocols is available in [4]) that exist on most networks in abundance, and are stored in very few places. These protocols define how information is communicated across networks; they provide a structure for network traffic (packets) which can be used to create a covert channel. This research focuses on the TCP and HTTP protocols. The viability of these two protocols as carriers is examined, and two different methods are compared.

The first method hides messages in a little-used portion of the TCP packet header; the second method hides messages inside of the data segments of HTTP packets. It is important to note that HTTP packets are TCP packets with specially formatted data segments, and that most networks handle significant amounts of this type of traffic. This allows these methods some degree of safety in numbers, as it increases the amount of overhead for automated controls. The comparison of the methods is based on how easily covert messages from Alice to Bob can be detected by Eve (a sound steganographic technique would be undetectable). Eve uses both human and automated detection techniques.

In terms of comparison, it is most important that the method be undetectable [5, 6]; should both methods pass detection testing, the one providing higher bandwidth (faster communication) would be more desirable. We believe that hiding messages in TCP packet headers will be more difficult for humans to detect, but will be more likely to trip an intrusion detection system (IDS) than the HTTP method. Hiding messages in the data of HTTP packets will allow a higher bandwidth than hiding in packet headers.

2. BACKGROUND

The need for a covert and secure communications channel forces us to use tools such as steganography (covert) and cryptography (secure). Steganography is a subset of the *information hiding* discipline that specifically covers the transmission of hidden messages. Cryptography, a much more developed field of study, centers on making a message unreadable to anyone except the intended recipient. In an actual implementation of a secure covert channel, steganography and cryptography should be layered together (For more on this topic, please see [7, 8]); however, cryptography is a far more advanced science at the time of this writing, our research will focus on the use of steganography to establish covert channels of communication.

The most well known application of steganography is probably *digital watermarking* [9]. Digital watermarking is the process of inserting a digital signal or pattern into a digital file, and its purpose is to guarantee the authenticity, quality, ownership, and source of the file. There are two general types of watermarking, overt and covert, and both use some degree of steganography. Obviously, covert (otherwise referred to as invisible) watermarks use some form of steganography to render themselves invisible. Overt watermarks use steganography to protect themselves from removal. The explosion of research in this application of steganography can be linked to the increasing need for digital rights management [9]. Due to this demand, most of the research in steganography and steganalysis is done with media such as photographs, music files, and software [1, 2, 10]. Indeed, the majority of Cole"s steganography book [3] focuses on them; only one chapter discusses the use of volatile media.

The process of inserting a steganographic message into non-volatile media varies based upon the medium itself. Generally speaking, sound and image steganography is based on the replacement of least significant bits [3, 10, 11], while text-based steganography varies widely based upon the intended use of the file [3, 9]. For example, published documents may have several lines shifted up one pixel, with each shifted line representing a set bit. Source code, on the other hand, may include a set of unimportant string variables that contain steganographic information ([3] has many good examples for different types of media).

There are many professional, commercially available steganographic message-hiding applications designed for non-volatile media such as JSteg, for JPEG images, and Invisible Secrets, which can handle JPEG, BMP, HTML, PNG, and WAV files (These and other programs are discussed at length in [3]). Research has resulted in strengthened algorithms, although most still focus on the models discussed above. Significant strides have been made in the analysis area, however. Steganalysis attempts are considered passive attacks on steganography, similar to the relation between cryptanalysis and cryptography. Two separate detection methods of steganalysis have evolved. Meta-detection methods are based on higher order statistics, and are capable of detecting messages inserted using any algorithm, although they must be „trained" on sample data for that medium. The method shown by Chandramouli and Memon [1] is one such attack, and is capable of 64% detection with a 2% false positive/false negative rate. Focused methods, on the other hand, attack a particular algorithm and medium, but are capable of near-perfect detection rates; the method presented by Zang and Ping in [2] is capable of 100% detection (false ±5%) against JSteg-like algorithms. There have not been any proposals for focused methods targeting algorithms used in volatile media-based steganography.

Unfortunately, there has been little formal research in the volatile media area, nor towards applying new steganographic techniques to network traffic, such as TCP, ICMP, and HTTP packets. In fact, the software tools used in this research were only developed as proof-of-concept.

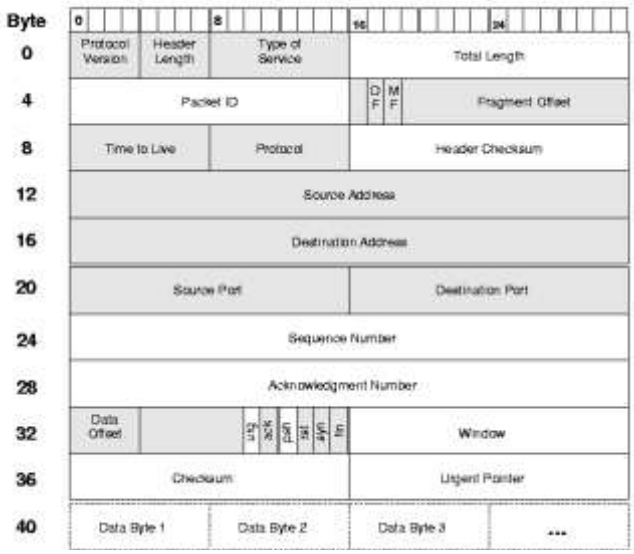


Figure 2. TCP/IP Packet Header Structure

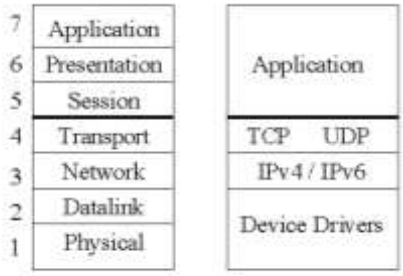


Figure 3. OSI Layer model (left) vs. IP Suite model (right)

3. SOFTWARE TOOLS

For this project, we located two steganographic tools and two detection tools. Our first steganographic tool, Covert-TCP, based on some of the theories presented in [11], works by embedding the message in one of two fields in the TCP/IP header. Fig. 2 displays the TCP/IP header; the first 20 (0 through 19) bytes are the IP header which is wrapped around the TCP header, bytes 20 through 40. One of Covert-TCP"s modes of operation embeds the message into the Identification field (bytes 4 & 5) of the IP header, the byte of message data is written into the first byte of the field, and the remainder zeroed. The second hides the message in the sequence number (bytes 24-27) of the TCP header; again, the byte of message data is written into the first byte of the field, and the remainder zeroed. A third method, involving hiding in the acknowledgment field (bytes 28 – 31) which was unused in our work, is mainly used when bouncing messages through HTTP proxies. Covert-TCP is specifically designed to transmit files. Our second steganographic tool is HTTPTunnel, an evolution of the ReverseWWW Shell discussed in Cole"s book [3], which is specifically oriented towards (remote) piercing of restrictive firewalls. It works by embedding the message in the data section (bottommost section of Figure 2) and formatting it to look like an HTTP 'get' request by adding the appropriate data to the beginning of the message. Port forwarding in the software allows the traffic to travel to and from standard HTTP ports, creating a tunnel. HTTPTunnel is designed to tunnel communications between the two machines, and is generally used in conjunction with telnet or secure shell. Both methods forge the packets used in the communications, meaning that they do not rely on the operating system to send the message for them; Covert TCP would not be able to modify the fields it does if it did not forge packets, nor would HTTPTunnel be able to spoof its source address.

The two detection tools acquired provide for human and automated analysis, respectively. The Advanced Packet Sniffer (APS) allows a user to filter sniffed packets by IP, protocol, hardware address, and others. It is also capable of displaying the data segment in hex, ASCII, or both. This operates on the transport and network layers of the OSI model (Figure 3). Snort is a popular open-source, lightweight intrusion detection system (IDS). It is a network based system capable of real-time traffic analysis and packet logging. Snort will be monitoring packet traffic to ensure that our methods do not draw attention to themselves and that an automated control does not detect their use. The major concern is the use of forged packets, something done by many of the attacks Snort has signatures for; if Snort flags this traffic, it has not only distinguished it from other traffic, but has also drawn attention to it, which would generally lead to suspicion and human analysis.

Finally, it is important to note that there is another form of automated control that was not a part of the research – a firewall. Firewalls come in two general forms, a packet-filter firewall or a proxy (application gateway) firewall [4]. Packet-filter firewalls are concerned only concerned with a small portion of the transport layer, which protocol is being used, the source port and address, and the destination port and address. Since neither of our methods utilizes these portions of the header, a filter firewall will have no impact on the results. Also, Snort monitors these fields [12], so abnormal values created by packet forging will still raise an alert; a filter firewall would block these packets, eliminating our ability to sniff and log them for human analysis. On the other hand, a proxy firewall, operating on the IP suite"s application layer (OSI layers 5-7, Figure 3), could be modified to specifically look for steganographic packets. However, there are no tools currently available to accomplish this, and our research does not focus on developing new detection mechanisms; therefore this approach was beyond the scope of our project.

4. METHODS

Our research methodology is broken down into four phases, in chronological order: Preliminary research, experiment mock-up, experimentation, and results analysis.

4.1 Environment Set-up

The basic computing environment for this experiment was defined to include: Linux Mandrake 8.2, with the following packages: Network Client, Network Server, Development, and Configuration. For Alice and Bob, the environment also included Covert-TCP, and HTTPTunnel. For Eve, it included APS, and Snort (which also required flex, bison, and tcpdump). The computers were connected on a shared media (hub based) network.

		Covert-TCP				HTTP Tunnel		
		Trial 1	*Trial 2*	*Trial 3*		*Trial 1*	*Trial 2*	*Trial 3*
Packet Detections		4	6	5		8	16	24
False Positives (FP)		75	88	59		9	15	19
False Negatives (FN)		245	244	246		2	4	6
IDS Triggers		0	0	0		0	0	0
Transmission Time		20 min	45 min	90 min		<1 sec	<1 sec	<1 sec
Detection Percentage:		2%				80%		
FP / FN Rates		30% / 98%				76% / 20%		

Figure 4. Detection Experiment Results Summary

4.2 Experiment Mock-Up

The next phase of the project was to fully set up the environment and ensure that the software was working properly. In order to do this, test runs of the experiment were conducted. Communications were initiated from Alice to Bob using a variety of methods. Non-steganographic communication included web browsing, telnet and secure shell; steganographic communication utilized both Covert-TCP and HTTPTunnel (telnet and secure shell tunneling). Eve was running Snort and using APS to log packets, which were lightly examined to see if the two traffic types were distinguishable. These trials verified that Alice and Bob were capable of communication via covert and standard channels, Eve was capable of listening in on those communications, and their use did not set off the intrusion detection system. Also, they indicated that determining the difference between steganographic traffic and standard traffic was not a trivial matter, which meant the full experiment could proceed.

4.3 Experimentation

Covert TCP and Reverse HTTP Tunnel are designed for different purposes. Covert TCP is designed to facilitate file transfers, while HTTP Tunnel's purpose is to provide a remote shell to the user. In order to have a basis for comparing these disparate techniques, a method to have them steganographically hide the same message was required. Since HTTP Tunnel is capable of routing hidden messages to daemons running on the host machine, our setup used telnet to conduct psuedo file transfers through the tunnel provided by the application. We used the *cat* command, which caused the remote machine to send the file through the tunnel and across the network, to the local machine. In this way, we were able to send the same file from Alice to Bob using both TCP/IP header and HTTP request hiding methods.

Our packet sniffing software was capable of dumping output in both hex and ASCII formats, so we decided to use text files for testing our transmissions, in order to provide a format consistent with a message and to facilitate detection by humans. The files were of varying sizes, and sizes were increased in each new trial, (1KB, 2.5KB, 5KB) as larger hidden file sizes play a large role in the detectability of the message when using other techniques [1, 2, 3, 10q]. However, since both of our programs break up files into smaller pieces (Covert-TCP: 1 byte/packet, HTTP Tunnel: about 1 kilobye/packet), file size will have little effect on a packet by packet steganalysis, since each packet will be „filled." Each file was transmitted twice per trial with Covert TCP and HTTP Tunnel.

We conducted testing in three trials, one for each test file of 1KB, 2.5KB, and 5KB, with three modes of transmission: no steganographic traffic (control group), all steganographic traffic, and mixed mode, consisting of both steganographic and non-steganographic traffic. As these transmissions occurred, Eve's packet sniffer (APS) recorded all traffic generated by Alice's IP that is destined for Bob's IP address. Also, the network's IDS (Snort) was monitoring traffic for attack signatures using its default configuration; this acted as an automated detection technique, searching for forged packets.

4.4 Results Analysis

Human analysis was done afterwards on the packet sniffer's logs. A group of 10 people with a background in computer science, networking, or both were allowed to examine the logs, which consisted of a header dump (with fields tagged), and a data dump displayed as hex values and ACII characters. These logs were somewhat distilled, since the original logs were far too long for human analysis in a reasonalbe period of time; the logs presented to the analysts were about 250 packets long, with about 80 steganographic packets. This ratio of 32% steganographic traffic was higher than in our original logs (about 10%), both of which are higher than they would be in an uncontrolled environment. Analysts were instructed in the general operation of both Covert-TCP and HTTP Tunnel, and asked if they could determine which packets contained hidden messages. We recorded their correct guesses, as well as false positives (analyst guessed that a packet contained a message when it did not) and false negatives (analyst guesses that a packet did not contain a message when it did). We complied detection, false positive, and false negative percentages for comparison.

5. RESULTS

As Figure 4 clearly demonstrates, the detection methods available to us were insufficient to detect messages hidden with Covert-TCP. Even with a dedicated attack, we were unable to ascertain the message when hidden in the IP Identification field. Covert-TCP proved to be far more difficult to detect than HTTP Tunnel, although very slow. HTTP Tunnel, on the other hand, transmitted the file in a matter of seconds. Unfortunately, we found that

human analyses of its HTTP requests returned a surprisingly high number of detections.

Only one of our analysts was able to determine a difference in the traffic generated by Covert-TCP, based on the zeroed end byte of the IP Identification field (he only detected about 20% of the message). HTTP Tunnel"s messages were only missed by 2 of our analysts, and the others had no problem reciting significant portions of the transmitted message. The fact that the former method sends the message one character at a time (versus the latter sending most, if not all, of the message at once) plays a large role in detectability; this result falls directly in line with previous research. However, the sizes of the files themselves had little to no effect on the detectability from trial to trial; it seems our analysts either broke the method or did not. Also, traffic analysis would be far more likely to identify the header method than HTTP method; One packet is sent about every second until the message is transmitted (for our files, this lasted about 20, 45, and 90 minutes apiece, respectively). HTTP Tunnel, however, sent only a few packets (1, 2, and 4 respectively) for each file.

6. CONCLUSIONS

As stated before, the most important factor in establishing a secure covert communications channel is that it be resistant to steganalysis; clearly Covert-TCP"s packet header hiding technique is the better choice with a 2% detection rate. However, HTTP Tunnel was not developed for that purpose, since it is directed at firewall penetration and often used in conjunction with an encryption technique like secure shell – in that scenario, it may be secure *enough*, if one is only concerned with fooling automated controls. Some of the contributing factors, such as the lack of a method to break up messages, or the limited use of HTTP formatting done to the text, could be corrected. It is also important to remember the much higher percentage of steganographic traffic in our tests – an uncontrolled environment would produce far more traffic that looked similar to the transmitted messages of HTTP Tunnel, especially if formatted with HTML tags. In fact, it is impressive that either method survived any dedicated detection at all, considering current success rates of dedicated detection techniques, and the relative immaturity of the algorithms used.

Future research could attempt to secure HTTP Tunnel"s transmission technique, although we think that may be difficult. As for Covert-TCP, its main problem is transmission speed. It currently only transmits information (either field) in one byte of the space designated to the field, so it could be increased to 6 bytes per packet (2 in IP Identification, 4 in TCP Sequence Number), although this may compromise its security. Future work could focus on a method to increase bandwidth and provide even greater security. Other studies could address some of the issues that we could not, for example, the use of application proxies to detect steganographic network traffic and developing an automated packet steganalysis tool.

7. ACKNOWLEDGEMENTS

I would like to thank Dr. Debnath, Dr. Francioni, and Dr. Smith for their continued guidance throughout this project, which helped me to focus its direction and scope, and their support in dealing with the details of the paper. I would also like to thank Dr. Cichanowski, without his networking expertise I would have been unable to learn as much about how these methods operate under the covers, and for his help with several sticking points. Also, Dave Hajoglou"s Linux and administration expertise was critical in the development of the experiment environment. I would like to thank the other students who helped to review this paper and refine its structure. Finally, I would like to thank my parents Dan and Maggie, because without their support I could never have gotten this far, and Bryanna, for putting up with me after consecutive nights without sleep. Thank you all! ☺

8. REFERENCES

1. Chandramouli, R. and Memon, Nasir. "A Distributed Detection Framework for Steganalysis," *Proceedings of the 2000 ACM workshops on Multimedia*, November 2000.
2. Zhang, Tao and Ping, Xijian. "A Fast and Effective Steganalytic Technique against JSteg-like Algorithms," *Proceedings of the 2003 ACM symposium on Applied computing*, March 2003.
3. Cole, Eric. *Hiding in Plain Sight: Steganography and the Art of Covert Communication*, Wiley Publishing, Inc. Indianapolis, IN. ©2003
4. Forouzan, Behrouz A. TCP/IP Protocol Suite, 2nd Edition, McGraw Hill Companies, Inc, New York, NY. ©2003.
5. Moskowitz, Ira, Chang, LiWu, and Newman, Richard. "Capacity is the Wrong Paradigm," *Proceedings of the 2002 ACM workshop on new security paradigms.*
6. Moskowitz, Ira, Longdon, Garth, and Chang, LiWu. "A New Paradigm Hidden in Steganography," *Proceedings of the 2002 ACM workshop on new security paradigms*, September 2002.
7. Desmedt, Yvo and Van Le, Tri. "Moire Cryptography," *Proceedings of the 7th ACM conference on Computer and communications security*, November 2000.
8. Rabinovich, Vlad. "Steganography - a Cryptography Layer," Available at: http://www.rit.edu/~vxr8205/crypto2/cryptopaper.html Accessed: Jan 2004.
9. Berghel, Hal, "Watermarking Cyberspace," *Communications of the ACM*, Volume 40 Issue 11, November 1997.
10. Kurak, C. and McHugh, J. "A Cautionary Note on Image Downgrading In Computer Security Applications," Available at: http://citeseer.nj.nec.com/cachedpage/14609/1 Accessed: Jan 2004.
11. Ahsan, Kamran and Kundur, Deepa, "Practical Internet Steganography: Data Hiding in IP," Avaiable at: http://ee.tamu.edu/~deepa/pdf/txsecwrksh03.pdf Accessed: Jan 2004.
12. Liang, Brian. "How to Guide-Implementing a Network Based Intrusion Detection System," Available at: http://www.snort.org/docs/iss-placement.pdf Accessed: Jan 2004.
13. Schneier, Bruce. *Secrets & Lies: Digital Security in a Networked World*, Wiley Publishing, Inc. New York, NY. ©2000.

9. APPENDIX A

Experiment Results Log

Analyst	TCP Method	HTTP Method		False P+ TCP	False N- TCP		False P+ HTTP	False N- HTTP
	t1 / t2 / t3	t1 / t2 / t3		t1 / t2 / t3	t1 / t2 / t3		t1 / t2 / t3	t1 / t2 / t3
1	0 - 0 - 0	1 - 2 - 3		2 - 3 - 0	25 - 25 - 25		1 - 0 - 1	0 - 0 - 0
2	0 - 0 - 0	1 - 2 - 3		5 - 2 - 3	25 - 25 - 25		1 - 3 - 0	0 - 0 - 0
3	4 - 6 - 5	1 - 2 - 3		0 - 0 - 0	20 - 19 - 21		0 - 0 - 1	0 - 0 - 0
4	0 - 0 - 0	0 - 0 - 0		5 - 10 - 5	25 - 25 - 25		0 - 1 - 0	1 - 2 - 3
5	0 - 0 - 0	1 - 2 - 3		10 - 14 - 7	25 - 25 - 25		1 - 4 - 2	0 - 0 - 0
6	0 - 0 - 0	1 - 2 - 3		6 - 4 - 8	25 - 25 - 25		0 - 0 - 0	0 - 0 - 0
7	0 - 0 - 0	1 - 2 - 3		12 - 22 - 16	25 - 25 - 25		2 - 4 - 7	0 - 0 - 0
8	0 - 0 - 0	1 - 2 - 3		23 - 18 - 9	25 - 25 - 25		1 - 1 - 4	0 - 0 - 0
9	0 - 0 - 0	1 - 2 - 3		12 - 8 - 4	25 - 25 - 25		0 - 0 - 0	0 - 0 - 0
10	0 - 0 - 0	0 - 0 - 0		0 - 7 - 5	25 - 25 - 25		3 - 2 - 4	1 - 2 - 3
Total	4 - 6 - 5	8 - 16 - 24		75 - 88 - 59	245 - 244 - 246		9 - 15 - 19	2 - 4 - 6
Out of	250 - 250 - 250	10 - 20 - 30		250 - 250 - 250	250 - 250 - 250		10 - 20 - 30	10 - 20 - 30
Percentage (%)	1.6 - 2.4 - 2.0	80 - 80 - 80		30 - 35.2 - 23.6	98 - 97.6 - 98.4		90 - 75 - 63.3	20 - 20 - 20
Average Percentage (%)	2	80		29.6	98		76.1	20
IDS Alerts	0 - 0 - 0	0 - 0 - 0	0%					

Evaluation of TRM Digital Fingerprinting Technology

Kevin Skerrett
Winona State University
Department of Computer Science
KMSkerre2200@webmail.winona.edu

ABSTRACT
A completely effective scheme for digital copyright protection is likely impossible. Files on a computer are easily copied. More significantly, they are easy to distribute. The need for functional protection of copyrighted materials is in high demand by those holding the copyrights. Digital Fingerprinting can be a first step in developing this protection without infringing on user's rights. With digital fingerprinting one can use the acoustic properties of a piece of music represented in digital form to identify the media file correctly. Using a program developed by MusicBrainz that utilizes a fingerprinting system TRM, developed by Relatable, we can effectively identify music media files using acoustic properties. The evaluation done in this paper will suggest that the TRM technology is an effective first step in the realm of digital rights management.

Categories and Subject Descriptors
D.2.9 [**Software Engineering**]: Management – *Copyrights*

General Terms
Theory, Legal Aspects, Verification.

Keywords
Digital fingerprinting, metadata, music downloading, digital copyright, acoustic properties.

1. INTRODUCTION

Currently on home computers it is easy to violate copyright laws by pirating media and using it in the comfort of your own home. Development of standards for protecting digital material has been slower than other forms of media due to the complexity of digital media laws and the difficulties of regulating information that is so easily copied and distributed. Because of this, copyright owners are increasing their demand for working, implemented digital rights management (DRM) systems to be used to protect their media [8]. However, copyright law itself is very difficult to interpret and is very context laden [9]. The public also does not want its personal computers to be locked down due to over zealous copyright protections. These are all factors that must be carefully weighed and balanced while considering a DRM scheme to protect the rights of the copyright holders.

One focus in DRM systems is the fair use of music files (specifically in the MP3 format). This is a very controversial form of copyright infringement of digital media due the ease of pirating (a full-song MP3 can be made to be an easily-transferable 2 to 5 megabytes) and its widespread use. It is clear that new techniques will have to be developed to implement an effective DRM solution.

The ramifications of these techniques should be explored. First It is important that these techniques are not easily exploited by workarounds, thus rendering the technology useless. Secondly The techniques implemented should not infringe on user's existing rights, such as fair use laws. Thirdly they must not adversely affect user's operation of their home computer by restricting digital content that should not be restricted.

The use of digital fingerprinting can be used to accurately identify a song and "tag" it with correct file information with a high degree of success. Digital fingerprinting is defined as generating a unique identifier to a file by its digital properties such as file size or composition. This concept would give a solid base for a built-in program to authenticate the media's information with a digital license server.

A typical DRM reference model works by transferring a policy (defined by what a user can do with the media) from some authority (the license server) to an enforcing agent closer to the user [9]. With this model in mind, a scheme must be developed to identify the media that is being used. A system that could use existing media formats and use its acoustic properties (an unchangeable property of a song) to do the identification would logically be effective and desirable. The Relatable Company has developed such an identification tool, called *TRM*, which works with the MP3 format.

This paper will evaluate the TRM technology. Background information regarding this classification of DRM system will be presented. MP3 metadata will also be discussed. Finally, we will show that this technology can identify MP3s effectively as demonstrated in a test of various MP3s. This successful technology is a first step in developing a reliable client side digital right management system.

Proceedings of the 4th Winona Computer Science Undergraduate Research Seminar, April 20–21, 2004, Winona, MN, US, Copyright 2004.

2. BACKGROUND

The idea of any copy protection scheme is that the users of the particular media should be able to use the media in an appropriate way. In this context "use" refers to playback, copying, or re-distribution. At the same time, the media should not be used inappropriately (including use of the media without first paying the publisher). In the digital realm this is a very difficult task. Some research even suggests that it may be impossible [3]. However, this is not stopping many firms, both national and international, on working towards a number of standardized DRM systems, as well as many proprietary DRM systems that are currently competing in the marketplace [11]. With this work comes some description of digital rights management, as well as a subset, Copy Protection and Content Management (CPCM), which is how digital fingerprinting is classified

A DRM system would be defined as having the following elements [11].

- Rights description – states how, where, when, and by whom the media can be used.
- Access/Copy control – means of preventing unauthorized use.
- Billing and payment systems – interface for collecting payment for use of the media.
- Identification and tracing – the means of identifying media and determining the usage.

CPCM systems share with DCM systems three foundation technologies:

- Encryption/scrambling: Process of rendering information unintelligible except to those in possession of keys needed to reverse process.
- Watermarking and fingerprinting: Watermarking is the hiding of a message within the media itself. Fingerprinting is the process of generating a message that is unique to the particular media.
- Authentication and identification: process of verifying that the media and other entities are who/what they say they are.

In the context of this work, the MusicBrainz project (which utilizes the TRM identification scheme) falls into the category of an incomplete CPCM system. It does nothing with encryption, and it does not watermark the media. It also has no way of controlling the person who is using the media.

3. IDENTIFICATION TOOLS

In order to identify music files, a scheme must be used that will provide an accurate and reliable way to distinguish the file from any other media file. Once that is accomplished, the metadata (see section 4 for more information on metadata) containing the distinguishing information needs to be stored and retrieved.

3.1 TRM Identification

Digital fingerprinting is the use of creating a unique identifier based on analysis of the media itself. This is focused largely, but not exclusively, on acoustic cues from music. TRM identification program [6] developed by Relatable can generate a unique 512 bit "tag" that is based on these criteria.

3.2 MusicBrainz Software

The open source project MusicBrainz [4] is an attempt to use the TRM tag described to uniquely identify the content of a particular media (in this case what song is encoded in MP3 format) and match the tag to a tag found on a central server. The tag on the central server is associated with information about the song it was derived from, including song title, artist, album and track. It is possible to have more than one tag associated with this information, as differences in how the music file was encoded can lead to different tags. When the tag is processed the server returns the closest matches, and the client software receives it and associated the information with the music file it originally scanned. If there is more than one potential match, the software allows the user to manually select the match he or she feels is the correct choice. For this reason the software is referred to as a "tagger." The MusicBrainz software then renames the media to the correct title, and adds the information to the metadata fields in the MP3. Everything about the MusicBrainz software is open-source except the actual TRM identification program. However, the TRM identification program can be used as a "black box" that will output a correct TRM identification tag when valid information is input.

MusicBrainz will run the song through the TRM identification program and retrieve a TRM ID [10]. That ID is compared against a collection of known TRM IDs that correspond to different media files. If a match is found, that match is returned to the user. If no TRM match is found a search based on the metadata from the file is carried out. The client (MusicBrainz) then compares the server suggestion with the metadata in the local file to return a percent similarity rating. This rating allows the user to receive feedback on the match.

The TRM signature that is returned will not necessarily be the exact same for each song. How the MP3 is created, any noise that may be present, silence on the lead in and lead out, or if the song is cut short by even a second will change how the tag is generated. Thus, any identification that is made is essentially a best-guess scenario by the client (MusicBrainz). However this paper will show that this guess can be made in a very accurate manner.

MusicBrainz developed as a tool for users to use and catalog their music library in a correct and uniform fashion [4]. If the percent similarity rating described above finds more than one match that rates very closely, the client will mark the song as "unidentified" and rely on the user to manually look at the matches and decide which one is correct. The choices are laid out in descending order by percent similarity. The most common instance of a song being marked "unidentified" is when the song is released in several different albums. Although acoustically the two (or more) songs are exactly the same the database considers the two (or more) to be different songs.

A database is maintained to store the metadata and TRM tags on the server-side. This database is user built. If a song that is owned by the user is not identified by the server, the user can manually input the information and upload it to the server. The existing server that MusicBrainz uses is very expansive, but it is nowhere near complete.

4. CURRENT MP3 METADATA

The metadata fields on MP3's come in two different types [5]. There is the ID3v1.1 standard, and the upcoming ID3v2 standard. All of the information encapsulated within ID3v1.1 is also encapsulated within ID3v2. ID3v2 has the same information fields as ID3v1.1, as well as additional fields unique to ID3v2. Thus you can convert from ID3V1.1 to ID3v2 with no data loss, or convert from ID3v2 to ID3v1.1 with the additional field information lost. For the purposes of this paper, we will be looking at the ID3v1.1 standard.

The metadata within an MP3 is stored at the end of the file [5]. This is to make it less likely to affect a decoder. The reason these metadata fields were placed in the file at all was to include information such as the title of the song, who the artist is, the album the song came from, what year it came from, etc. It was chosen that the size would 128 bytes, and the data was arranged as shown in figure 1 [5].

Song title	30 characters
Artist	30 characters
Album	30 characters
Year	4 characters
Comment	30 characters
Genre	1 byte

Figure 1: MP3 Metadata

5. METHODS

To give an indication on the effectiveness of the TRM identification program a set of test data was ran. The intent of the experiment is to suggest that the TRM identification process is an effective solution to identifying MP3 media.

5.1. Process Setup

The process required setup ahead of time before the actual experiment could take place. A set of 75 music files were used for the tests. It was necessary to ensure that the 70 "identifiable" songs existed in the user-built database, and 5 were not. This step is appropriate for the test because it is safe to assume that any media that needs to be protected from copyright infringement should be well documented on the server side of a DRM solution. The 5 other songs are used to ensure that songs cannot be identified are not misidentified. For comparison purposes two copies of the same song that came from different sources was included. We will refer to these files as the "twin" files.

60 of the identifiable songs came from various artists, and all contain vocals and drum beats. 10 of the songs are instrumentals, 5 of which being considered classical.

5.2. Interface

The interface of the Music Brainz software is GUI interface that supports "drag and drop" features within the operating system. Currently the software is only available on a Microsoft Windows platform (see appendix a screenshot of the user interface). The software organizes the media into a handful of categories:

- *Unidentified*: This category contains files that cannot be identified conclusively by the software. It may have too many possible matches or there may not be any suitable matches.
- *Identified:* This category contains files that have been identified by the software. A percent similarity rating is assigned.
- *Pending:* This category contains media that is sitting in the queue waiting to be analyzed.
- *Saved:* This category contains files that are saved by the user. The files in this section have had their metadata and filename changed to the values that have been supplied by the MusicBrainz server.
- *Error:* This category contains files that were unable to be completed. This is usually due to an inability to communicate with the server or a timeout. The files can be moved from here to the pending category.

5.3. First Test: Accurate, Incomplete Metadata

The first test was done to represent a real-world implementation of the digital fingerprinting and matching process. The importance of this test was to show how the software will typically behave and is not meant to test the boundaries of its capabilities. The term metadata refers to the filename as well as the ID3v1.1 and ID3v2 tags on the MP3s.

70 songs were chosen and the metadata on each was accurate, though incomplete. The filenames contained the song title in some cases, and the song and author in others. The metadata

similarly contained a combination of author, title, etc. It is assumed that in a real world application the files being examined would not have a uniform amount of metadata that will be filled in, thus in this experiment which particular files contained which metadata is unimportant.

All 70 songs were put into the "pending" category. Any songs that came up as "error" due to network difficulties were retried as many times as needed (typically no more than twice).

5.4 Second Test, No Metadata

In the second test all metadata was removed from the files. The songs were given the filenames 1 through 50 in no particular order. The new files with no metadata were examined the exact same way as the previous songs. The same care was given to any errors that may occur.

This test will give an indication as to the accuracy of the TRM technology. Since there is no metadata on the files, the tagger software will have to rely entirely on the acoustic properties to identify the files.

5.5 Third Test, Successful Failure

In the third and final test a small set of MP3s were used to test the effectiveness of the MusicBrainz program when tested with an MP3 file that does not show up in the database. For this I used 5 pieces from an unpublished and undistributed artist. The songs were not knowingly based on any other pieces and thus have a high probability of having a unique acoustic signature. The songs chosen were verified to not exist on the match server. A successful result will yield no known matches, which is correct.

The MP3s were given correct filenames, and correct metadata.

6. RESULTS AND ANALYSIS

6.1. Test 1: Accurate, incomplete metadata

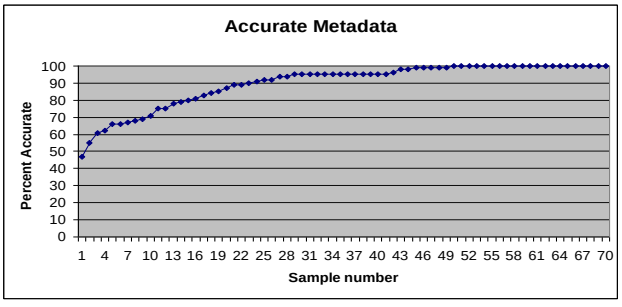


Figure 2: Result of first test

The results are shown on the scatter plot of figure 2. The vertical axis is the percent similarity rating that was associated with the corresponding MP3. The horizontal axis represents the sample number. To make it easier to read, the numbers were ordered by their percentile. The significant data in this test is how many songs were identified correctly and which were not. The results are as follows:

- Identified Files: 64
- Unidentified Files 6
- Total 70

This test gave a success rate of 91%. It should be noted that of the 6 unidentified files, 5 were only classified as "unidentified" by the MusicBrainz software because different collections containing the same song came up as close possibilities for the tagger. Since it found essentially the same song in all 5 cases it is arguable that the success rate for this run was 98%.

The remaining 1 file that was unidentified was a classical piece. The client had identified the song correctly but by the wrong composer.

The "twin" files resulted in data of note. One of them was assigned a percent similarity rating of 85%, and the other was one of the 6 unidentified files with a rating of 67%.

The 10 instrumental songs were marked considerably lower. Only 6 of the 10 were identified. The lower success rate is potentially due to the fact that it is more difficult for the TRM identification program to differentiate when the same classical piece is done by two different performers.

6.2. Test 2: No metadata

The results are shown figure 3. Please note the order of the songs across the horizontal axis is identical to the previous graph.

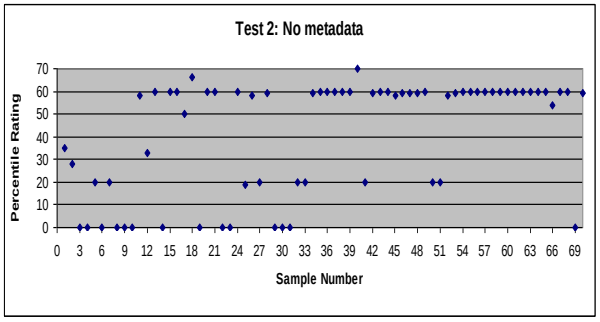


Figure 3: Results of second test

This graph has much higher success on the right side of the graph compared to the left side. This corresponds to the higher percent similarity ratings from the previous test. A percentile ranking of 70% is the highest rating that can be found without any metadata. The one song that was ranked at 70%, song #27, ranked the highest because the numeral 2 was in the actual song title as well as its filename, 27.

The success rate of the client can be measured by how many of the songs were identified correctly, i.e. songs that were automatically put into the "identified" category. Songs that were

unidentified but had the correct song as the first choice for the user to manually select would also be labeled as a success. As stated before, the songs to be identified were ordered by highest percent similarity rating. These results are as follows:

- Identified Files 64
- 1st Choice 38
- 2nd Choice 5
- No Good Match 14

For our test this gives us a success rate of 51 out of 70, or 73%.

The instrumentals once again had a lower rate of success. 5 of the 10 had no good match. Without the metadata it was impossible for MusicBrainz to determine which piece of music from which performer was being played.

6.3 Test 3: Successful Failure

The third test returned successful results. The MusicBrainz software attempted to identify all 5 songs and was not able to come up with conclusive results. The program returned that there was no match found for every song. This small-scale test suggests that the software is able to cope with unidentifiable media and not attempt to assign incorrect metadata to the media.

7. CONCLUSION

The results were positive. When stripped of all metadata the TRM identification program identified the songs with a high degree of success. When attempting to identify media that was not on the match server, the software did not incorrectly assign an inaccurate match. The results of this experiment suggested that it is possible to correctly identify media by its acoustic properties, which can be used as an integral part of a DRM scheme.

The database that was being drawn from did not contain many TRM IDs. If more known TRM IDs were available, it is reasonable to infer that the results would have been more successful. As it is, the database is user driven. Because of this there is not a uniform spread of TRM IDs. The more popular the MP3 is the more TRM IDs will show up in the database. This could explain the lower success rate in the instrumentals, as it is reasonable to assume these were not identified by the user base as often.

Currently Relatable is in a partnership with Napster to develop a system to identify the music files for their service.

8. ACKNOWLEDGEMENTS

I would like to give my thanks to Robert Kaye and Dr. Joan Francioni for their help on this project.

9. REFERENCES

[1] Clark, J. Accessibility implications of digital rights management. April 2003. http://www.joeclark.org/access/resources/DRM.html. Access on Jan 2004

[2] Russ, A. Digital Rights Management Overview. July 2001. http://www.sans.org/rr/papers/index.php?id=434 Access on Jan 2004

[3] Schnier, B. The Futility of Digital Copy Prevention. May 2001. http://www.schneier.com/crypto-gram-0105.html#3 Access Jan 2004

[4] Open MusicBrainz Project. MusicBrainz www.MusicBrainz.org Access: Jan 2004

[5] Nilsson, M. ID3 Made Easy. http://www.id3.org/ Access Jan 2004

[6] TRM Specification. Relatable. www.Relatable.com . Access Jan 2004

[7] Haitsma, J., Kalker, T. A Highly Robust Audio Fingerprinting System. April 2002 http://ismir2002.ismir.net/proceedings/02-FP04-2.pdf Accessed Jan 2004

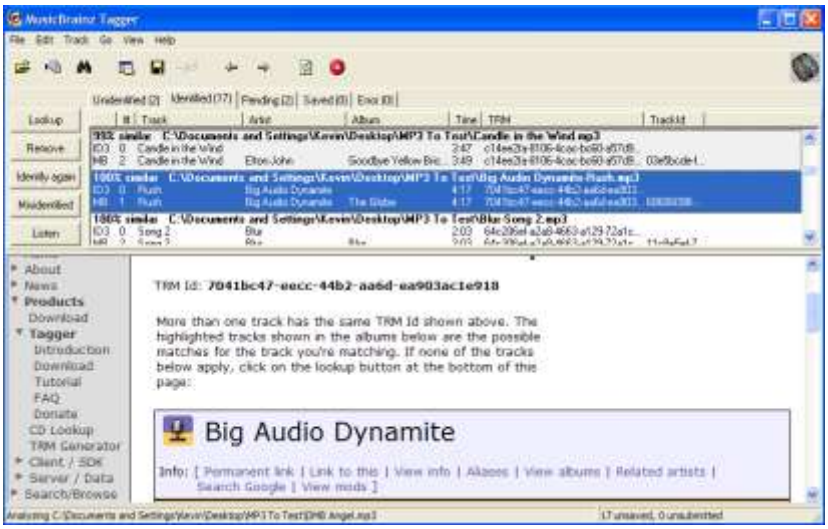
[8] Mulligan, Deirdre K. Digital Rights Management and Fair Use by Design. Communications of the ACM. April 2003, Vol. 46, No. 4.

[9] Erickson, John S. Fair Use, DRM, and Trusted Computing. Communications of the ACM. April 2003, Vol.46 No. 4.

[10] Kaye, Robert. Senior Developer of MusicBrainz. Correspondence via Email. February-March 2004.

[11] Vevers, R., Hibbert, C. Copy Protection and Content Management in the DVB. 2002 IBC Conference. September 2002

Appendix:



Screenshot of MusicBrainz Software

A Cost-Benefit Perspective of Grid Computing

Joseph Cropper
Computer Science Department
Winona State University
JWCroppe2030@webmail.winona.edu

ABSTRACT

Grid computing is represented by a collection of distributed computing resources available over a local or wide area network that appears to an end user or application as one large computing system. Several small and medium e-Businesses encounter peak times during which customer requests are not always met. Several industry leaders argue that grid computing offers affordable solutions to this problem. We show that the cost effectiveness of grid computing is largely dependent on the primary server's processor family and the size of the data transferred between grid nodes. More specifically, it is shown in one case that grid computing solutions offered nearly equivalent levels of throughput and a savings of over four thousand dollars when compared to the addition of a second processor. However, we also showed that the total savings decreased to merely a few hundred dollars and that overall grid computing throughput levels dropped by 20 percentage points as the size of the data to be replicated among grid nodes approached five hundred kilobytes per subtask.

General Terms

Management; Performance; Experimentation.

Keywords

Grid Computing; Grid Simulation; Performance Evaluation; Small and Medium e-Businesses; Scalability; Cost-Benefit Analysis; GridSim.

1. INTRODUCTION

Grid computing is making an appearance in the modern day e-Business world as a growing field of interest to researchers and industry leaders. The demand upon e-Businesses to provide high availability, high reliability and immediate responsiveness by their customers is growing rapidly in today's world. While there are certainly high-end servers that can handle a large, ongoing flux of user requests, many small and medium e-Businesses cannot afford them. In the context of small and medium e-Businesses, high-end servers include expensive symmetric multi-way processor architectures and machines of a much higher caliber, such as an IBM iSeries. However, through the benefits of a grid, companies are presented with a cost effective mechanism to meet more customer demands than with their standalone server, without having to upgrade to a considerably more expensive primary server, as mentioned above. Keep in mind that when the term "server" is used, it is not restricted to any one type of server (e.g., a web server, a database server, etc.), but rather any machine that provides one or more service to a set of clients.

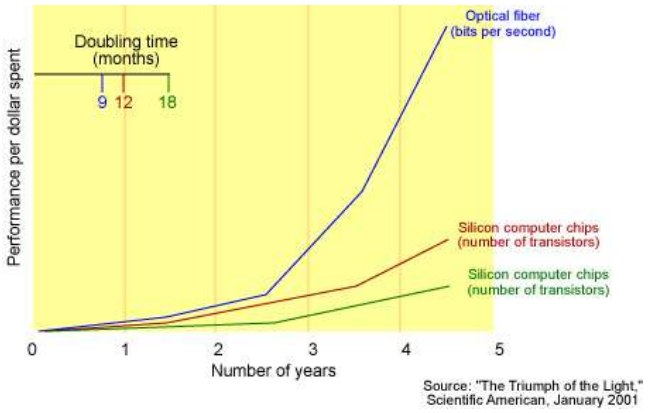


Figure 1. A chart showing Network and Processor advances.

Recall Moore's Law, which says that the number of transistors on a microchip doubles every 18 months, has a corollary: the costs of processors are dropping at rates of nearly 25% per year [1, 2]. This is important because it means that an e-Business can expand a grid by purchasing older machines at relatively low costs to rescue the primary server during peak times. During non-peak times, these extra resources can be used for general-purpose computing. Figure 1 shows the recent divergence between the rates at which networking capabilities increase relative to the number of transistors on an integrated circuit. To harness the capabilities of this recent advancement in networking, it is apparent that a more efficient way of harnessing processing capacity is necessary [2].

It is also known that distributing tasks among several resources can potentially increase overall system productivity. Today this is frequently performed in research facilities engaging in computationally intense tasks. Similarly, this concept can be applied in the e-Business domain, but present grid technologies are still premature for global open-standard industry use [3]. Grids must be easy to build and their services must be easy to utilize, which presently is not true. There must be open standards to which information technology professionals must adhere when

Proceedings of the 4th Winona Computer Science Undergraduate Research Seminar, April 20–21, 2004, Winona, MN, US, Copyright

designing and implementing grid computing solutions [4]. There must also be evidence that grids have the potential to benefit companies financially, which is one of the many sparse areas of grid computing studies.

This research focused on a simulation of a grid-enabled system that underwent various degrees of workloads. We examined how a grid benefited (or did not benefit) e-Businesses both financially and computationally. There are four possibilities from which companies can choose when deciding how to handle their periods of peak processor usage:

1. Do nothing and continue to use current resources.
2. Purchase another server with increased processing capabilities.
3. Instantiate a grid environment.
4. Lease processor bandwidth from another company whose resources meet the minimum requirements to fulfill the customers' requests.

The first option is acceptable for companies whose processor limitations are never exceeded. But what happens when these limitations are exceeded? When processors reach their point of peak computation, there is a window of exposure where customer requests may be lost or denied. This research compared the cost perspectives of the second and third options above and showed how instantiating a grid may or may not be a less costly solution to meeting more customer demands during occasional peak times, relative to upgrading to a more powerful primary server.

The environments simulated in the experiments deployed constant sets of resources, each of which were exposed to a varied number of users. There was also a wrapper experiment that re-examined the cost effectiveness of grid solutions as the amount of the data exchanged between nodes increased. While the simulator did not take into consideration the problems posed by network topology, these scenarios still provide e-Business executive and technical officers with a general understanding and a conceptualization of how various resource configurations will respond to different degrees of workloads. By finding a balanced medium between the costs of additional resources, the additional number of customer demands met, the size of data exchanged between nodes on a grid and the scalability of the solutions, executive decision makers are provided with some heuristics to innovate their e-Businesses.

2. BACKGROUND RESEARCH

2.1 What is a Grid?

There has been much confusion and debate within the technical community over a single definition of a grid. In a 1998 article, Foster said that he and Kesselman defined a computational grid as a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities [2]. Later in a 2000 article, Foster restricted his definition to systems that use standard, open, general-purpose protocols and interfaces that delivers non-trivial qualities of service [2]. Grid computing differs from clustered computing in that clustered computing resources generally exist on a single local area network, consist of a static set of resources and usually consist of a single type of operating system, whereas grid computing resources can potentially span several

Figure 2. A generic view of the World-Wide Grid Computing environment [2].

geographical domains via wide area networks, consist of a dynamic set of resources and contain heterogeneous systems [2].

The concept of re-assigning tasks to alternate resources to allocate resources for new incoming tasks is not a strikingly new concept. This idea of parallel computing is currently employed in several compute-intensive environments, such as university laboratories and research facilities, to increase overall system throughput. As availability of powerful computers increases and high-speed network costs decrease, researchers and industry leaders are searching for ways to incorporate and standardize the grid architecture for small and medium e-Business paradigms to increase the availability of their servers at an affordable cost [5].

As seen in Figure 2, users submit their jobs to a resource broker (which often runs on the primary server). It is a resource broker's job to determine where to send a user's request. Grid resource brokers receive topological information (i.e., which nodes are available and how they can be reached) from a grid information service directory. Now the grid brokers have enough information to decide on which node to send the job and where to deliver it (i.e., the node's IP address). Grid brokers base their decisions on some predefined scheduling policy. In the case of GridSim, a round-robin timeshared policy is employed.

Previous work in the domain of grid computing has largely been focused on how to effectively manage grid resources, schedule grid tasks, achieve fault tolerance, securely exchange data and how to assemble an open-standard protocol that defines communication between grid nodes [6]. Very little work has been done in terms of how grid computing financially benefits e-Businesses. Grid computing in the e-Business world has just recently become a focal point of attention to industry leaders [2]. Our research focused on specific examples of when and where grid computing would be applied in the e-Business world and how it benefits companies both financially and computationally.

2.2 GridSim: A Java-based Grid Simulator

Tools for evaluating the cost and performance of various grid configurations are necessary as the potential for grid computing solutions in the e-Business domain grows. Recognizing this potential, Dr. Rajkumar Buyya of the University of Melbourne devised the open source GridSim project. Started in 2000, the primary goal of the GridSim project is to provide the tools necessary to simulate potentially thousands of users with varied requirements and a large set of potentially heterogeneous

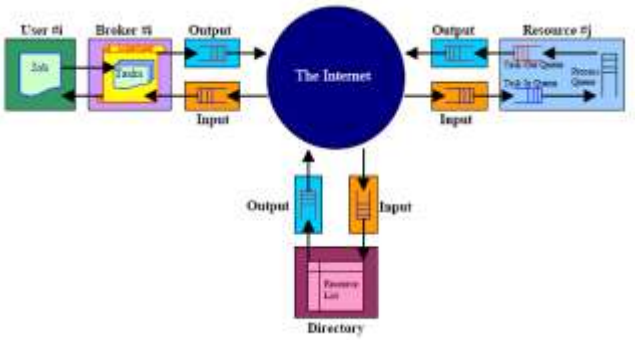


Figure 3. A flow diagram in GridSim-based simulations [3].

resources [5]. The simulator has undergone several revisions and currently provides options for customizing many parameters such as: the system architecture, task schedulers, processor specifications and network baud rates between nodes on a grid. These variables provide the users of GridSim with the flexibility to effectively evaluate a wide variety of grid computing properties. In the past, the simulator has typically been used to create and evaluate grid scheduling algorithms and now it is used to study the cost-oriented perspectives of grid computing [7].

Figure 3 shows an abstract view of GridSim's design. Similar to Figure 2, the user essentially interacts with a resource broker. The broker interacts with the directory to determine which resource should receive the user's jobs. Upon completion, a resource is returned to its broker and the user is notified that his or her job completed.

3. METHODS

3.1 Hypothetical e-Business

A hypothetical e-Business was devised to draw parallels between the raw results and what these results signify in the real-world. The environment simulated in the experiment represented that of a small to medium-sized e-Business that provides its customers with a dynamic web-based interface to find a place to live. The service allows any client to search a nation-wide database of places to live. It also allows subscribing landlords (i.e., the paying customer) to dynamically update their listed properties. When a landlord updates his or her properties, the process entails entering standard HTML form data and optionally submitting a set of images for each property. These images are usually in JPEG format and are approximately 40 KB in size. However, sometimes there is the occasional user that submits pictures of a higher quality, and consequently of a greater size. For user convenience, when an image is uploaded, it is copied and transformed to decrease the image to thumbnail (approximately $1/10^{th}$ of the original) size. The user can optionally click on an image to see the picture in its full size. Fees are based upon the number of properties that a landlord has listed. As a special promotional offer, there are select days during the year when landlords are offered a lower cost to list properties (i.e., the peak usage times). During these periods of time, the primary server experiences much more frequent property listings than usual because landlords are franticly listing properties. Due to several users submitting images more frequently than usual, the server fails to service some landlords' requests in a reasonable amount of time. Consequently, these landlords' web browsers timeout and the transactions are aborted.

3.2 Experiment Design

To examine the cost-benefit ratios of grid versus non-grid solutions, a simulation was carefully designed and executed. To model a typical small or medium business' resources, commonly purchased servers were selected for use in non-grid scenarios. Vendors of these resources included IBM, Hewlett-Packard (HP), Dell and Sun Microsystems. The idea was to expose this set of non-grid resources to a wide range of workloads, which is defined in the last paragraph of this section. Next, the same set of workloads was exposed to the same base set of non-grid resources, but with the caveat that each resource had been given increased processing capability. This modeled the company making the decision to upgrade their primary server to handle peak time workloads. Having done this, a chart depicting cost versus additional serviced requests was created. A grid solution (consisting of either a Pentium III or a Pentium 4) was also realized and exposed to the same set of workloads as previously mentioned. A cost versus additional serviced requests chart was created and compared with the chart described above.

Because the success of grid computing is largely dependent on the amount of data replicated during task migration, we devised a wrapper experiment varied the amount of transmitted data between nodes on the grid [7, 8]. Specifically, we experimented with three simulation data units per subtask: 364.08, 1820.44 and 4551.11. (Note that 1 simulation data unit mapped to approximately 112.5 bytes in the real-world). These simulation units mapped to real-world data sizes/subtask of: 40 KB, 200 KB and 500 KB, respectively. We also assumed that the grid nodes were connected via symmetric (i.e., the ability to upload and download at equivalent transfer rates) link speeds of 17476.26 simulation data units per unit of simulation time. This mapped to 512 Kbps in the real-world, about the speed of a fast broadband connection. The results of this experiment provide executive decision makers with the insight as to whether or not a grid will be beneficial to their company based on the expected size of the data to be transferred between grid nodes.

This simulation modeled a varied number of users, ranging in quantities between 1 and 500. Each user (i.e., a landlord) in the experiment submitted a job (i.e., listed a property) that consisted of 20 subtasks (i.e., submitted 20 images) that took approximately 0.80 units of processor time. For example, if the processor operated at 0.80 processor units per unit of simulation time, then the job would have theoretically completed in 1 unit of simulation time. To map simulation time into real time, it was assumed that 1 unit of simulation time equaled 30 seconds of real time, so the job would have taken 30 seconds to complete. (Keep in mind that this was just an arbitrary mapping and it was devised only to draw parallels between the simulation and the real-world.) However, as several users began frequently submitting jobs, this ideal completion time was no longer feasible. There was also the added restriction that each job must have completed within 10 units of simulation time (i.e., 5 minutes in real-time). Any subtask that had not completed within this timeframe was permanently marked as incomplete. This deadline is similar to the timeout value used in modern web browsers. Figure 4 shows the rate at which

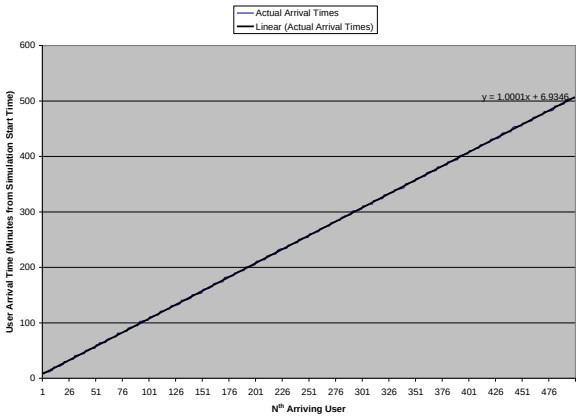


Figure 4. A graph of user arrival times.

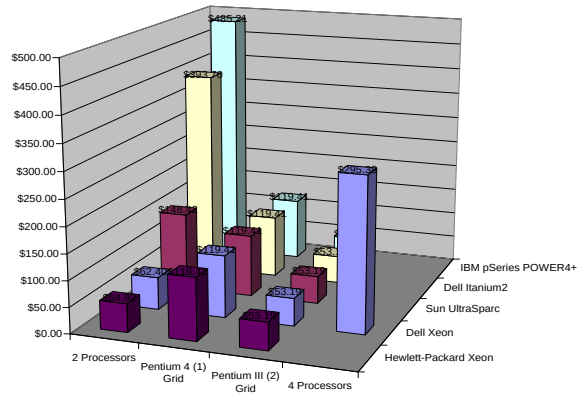


Figure 5. Cost per extra unit of processor.

users submitted jobs over an 8-hour period. There is a best-fit extrapolated line modeling the linear relationship between the n^{th} arriving user and the time at which that user arrived. Figure 4 shows that users arrived at an approximate rate of 1 user per minute. This submission rate was considered very abnormal (on the high end) to this hypothetical e-Business.

3.3 Processor Attributes

A critical component of the experiment was relating different resources' processing capabilities to one another. The GridSim toolkit accepts a parameter that defines the "speed" of a machine's processing entity (PE). There are two common ways in which the effectiveness of a PE is measured, either by its clock speed or by its MIPS (Million Instructions per Second) rating. These measurements work well when comparing architectures within the same family (i.e., Intel versus Intel or Sparc versus Sparc). However, when cross-architecture comparisons are solely based upon the clock speeds and the MIPS ratings, because of different instruction sets, very misleading results can be produced. To overcome this problem, the Standard Performance Corporation (SPEC) devised a set of benchmark tests to reliably compare and contrast a diverse set of resources' processor and memory capabilities [9]. Based on the results of these benchmarks, each participating machine was assigned a performance rating, which was used to relatively compare resources from different architecture families. To reliably compare and contrast the resources selected for this experiment, the SPEC CPU (INT) 2000 benchmarks were passed as arguments to each GridSim PE. While the SPEC measurements neither represented clock speed nor MIPS ratings, they still provided a mechanism to make relative statements about participant machines [10].

The cost of a computer is highly variable. With all of the online stores and e-Auction sites today, like e-Bay and Yahoo!, computer prices are constantly fluctuating. However, during this experiment, it was assumed that small and medium e-Businesses usually buy their servers from licensed vendors because they want extended manufacturing warranties, a luxury other purchasing options often do not have. With this assumption, the resource costs were queried from the vendors' online e-Stores. To stabilize the prices of the individual machines, the prices were queried twice per week over a four week period and no changes in price were observed. Each machine was priced with similar hardware specifications: 2 GB of RAM, 36 GB SCSI/10K RPM hard drives, no operating system and no display device. While there were certainly some variations in the cost of RAM and hard drives, these components could not be ignored. It was imperative that they co-existed to create a functional system, and therefore, they needed to be considered in the cumulative cost. Although computer prices are subject to frequent change and these results were only truly accurate as of this writing, there will be similar trends in pricing the server world versus the PC world, at least for the foreseeable future.

4. RESULTS AND ANALYSIS

4.1 Processor Costs

A key component to improving the overall throughput of jobs is adding more processing power. The cost of anything is important to any businessman or businesswoman. In this case, particular attention was given to the cost per additional unit of processor for grid versus the non-grid scenarios. The vendors of the modeled resources included Dell, HP, IBM and Sun Microsystems. Three different solutions to the hypothetical business' peak usage problem were studied. These solutions included upgrading to either a dual or a quad processor machine, a grid system with a Dell Precision Workstation 340 with a Pentium 4 processor and a Dell Precision Workstation 420 with dual Pentium III processors. In the case of the two grid scenarios, the original machines were left unaltered and the grid nodes were added. In the other two cases, upgrading the primary server to either dual or quad processors, the additional processors were purchased and installed separately. Figure 5 shows the breakdown of the results, categorized by vendor and system configuration. At nearly $500, IBM's POWER4+ dual processor architecture upgrade had an enormous cost per additional unit of processor ratio. The same applied to Dell's Itanium 2 dual processor architecture, at almost $400 per additional unit of processor. On the other hand, the Dell and the HP Xeon dual processor solutions both ranked in with cost-benefit ratios just under $75. This extreme difference in price was expected; Itanium 2 and POWER4+ processors are 64-bit, have larger resident caches and wider system buses than Xeon processors.

During the grid tests, the same two resource configurations were used, which explained the constant ratios for the Pentium 4 and

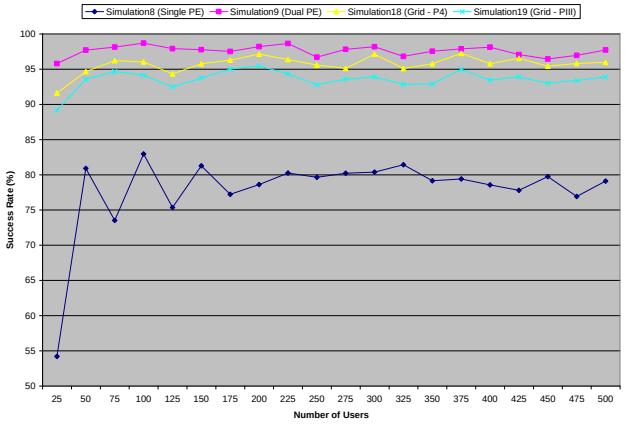


Figure 6. POWER4+ success rates with 40 KB/subtask.

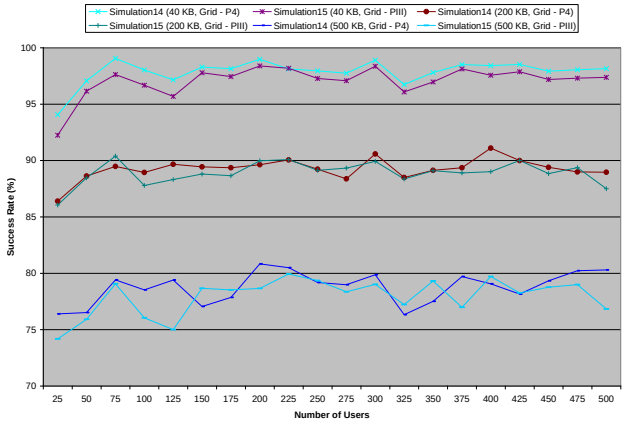


Figure 7. Dell Xeon grid success rates with varied sizes of replicated data among grid nodes.

dual processor Pentium III scenarios (Figure 5). At $119.41 (P4) and $53.19 (PIII) per additional processor unit, the Pentium grid solutions appeared preferable when compared to the Itanium 2, POWER4+ and Sun UltraSparc dual processor solutions. This is also true when compared to Dell's quad Xeon processor solution. However, the Pentium 4 grid solution did not stand out very well from the Xeon dual processor solution. Given that one could upgrade from a single to a dual Xeon processor for approximately $600 and gain 9.6 units of processing power (e.g., as seen in the Appendix during the transition from Simulation 3 to Simulation 4), the added $1409 11.8-unit Pentium 4 grid solution did not appear to be all that beneficial in terms of its added cost-benefit ratio. The dual processor Pentium III grid system ranked in at $53.19, merely $9.61 less than the dual Xeon processors. Although less expensive, as an initial upgrade, after the setup and the maintenance involved with grid computing, the trouble probably would not be worth the effort.

These results showed that the total savings in cost per additional unit of processor was more prominent when the server's primary processor was Itanium 2, POWER4+ or UltraSparc. As a first step, companies running Xeon technology should probably invest in a second processor before investing in grid technologies. Otherwise, it appeared that grid computing offered companies a significant savings (in terms of cost per extra unit of processor), all while achieving nearly equivalent levels of throughput.

4.2 Performance of Grid Solutions

The major components of the data collected during the simulations were the success measurements of the various resource vendors. Figure 6 shows the success rates of IBM's pSeries machine before any upgrades were made (i.e., moving to dual PEs or grid solutions), which explained the extremely poor success rates of the single PE. The chart shows that the proposed inexpensive grid solutions produced success rates nearly equivalent to the expensive dual POWER4+ processor solution. When used in correct environments, the cost effectiveness of grid computing definitely seemed to be evident. See simulations 8, 9, 18 and 19 in the appendix for pricing details. This success pattern was evident among all resource families. However, these grid success rates proved to be slightly misleading as the size of replicated data increased.

4.3 Effects of Data Replication

One of the key factors of the effectiveness of grid computing is minimizing the amount of data that needs to be replicated among nodes on a grid [7, 8]. Remember that a grid is transparent to an end user or application, and likewise, its expectations of the system do not change accordingly. That is, if an end user or application submits a job to a non-grid system and expects it to finish in *x* time units, that end user or application would also expect it to finish in *x* time units on a grid system. The time spent replicating data on a grid node is considered overhead time (i.e., time otherwise not spent during non-grid scenarios).

There was a definite trend in the relationship between the success rate of jobs and the size of data replicated among grid nodes. The job success rate was inversely proportional to the size of the replicated data. That is, as the size of the data increased, the success rate decreased. This relationship was evident among all of the simulations. Figure 7 shows how the various data sizes affected the Dell Xeon grid resource configurations. Transitions in the size of replicated data from 40 KB/subtask to 500 KB/subtask caused the success rates to drop by an average of 20 percentage points among all of the simulated resource configurations. Because a subset of the images were transferred over the grid and images $1/10^{th}$ of the original sizes were sent back to the primary server, the cumulative size of the data transfers ranged from 44 KB to 563.2 KB. This large drop in success rates also put a burden on the cost-benefit ratios of grid solutions. We observed a proportional relationship between the cost per added percent success and the size of the replicated data. As the replicated data sizes increased, the cost per added percentage point in success increased. Figure 8 shows the cost-benefit ratios for replicated data sizes of 40 KB/subtask among all resources. It is shown that Pentium III grid solutions had lower costs relative to upgrading to dual processors. All grid solutions proved to be less costly purchases when compared to the POWER4+, the Itanium 2 and the quad Xeon processor families. However, as the size of the replicated data increased, the cost-benefit ratio ratios decreased. Figure 9 is similar to Figure 8, but Figure 9 shows the success rates with 500 KB/subtask of replicated data. While the Pentium III grid solutions still seemed to be the best option in terms of cost, when the primary server was composed of either Itanium 2 or quad Xeon processors, this was no longer true for both the Xeon and POWER4+ processor

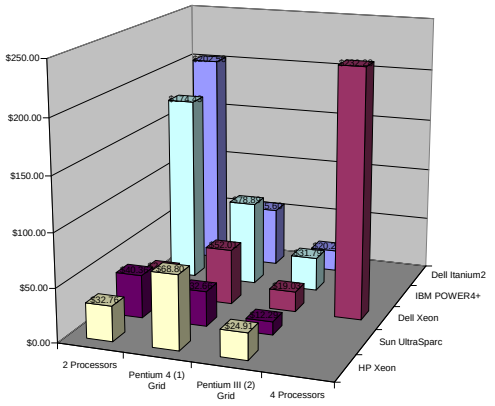


Figure 8. Cost per added percent success with 40 KB/subtask.

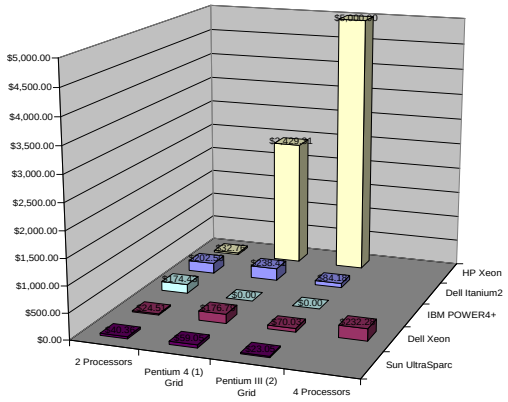


Figure 9. Cost per added percent success with 500 KB/subtask.

families. The 500 KB/subtask grid solutions had performance ratings on average 20 percentage points less than the 40 KB/subtask grid solutions and had higher costs per added percent success relative to adding a second processor. Because no data transfers were required (i.e., overhead), the second processor options did not experience any performance degradations. Not pictured, the 200 KB/subtask experiment produced ratios that fell between the 40 KB/subtask and the 500 KB/subtask replicated data sizes. Even though some grid solutions had an apparent cost advantage over the dual processor solutions, as the data sizes increased, the success rates fell far below acceptable levels. As seen in the case of HP Xeon, the extremely high ratios are a result of grid success rates being less than one percentage point better than the dual processor solution. These experiments showed that the cost effectiveness of grid computing relied on more than just processor capability, but also on the amount of data to be replicated and link speeds that connected grid nodes.

4.4 Total Financial Savings

While e-Businesses are certainly concerned with how much bang for the buck they are getting (i.e., the percentage point increase in serviced requests per dollar spent), they are especially concerned with the cumulative costs for the solutions to their problems. When 29 enterprise technologists were asked if they were using or evaluating grid solutions, 69% responded that reducing overall capital costs was of most importance [2]. Simply because one solution yields a higher ratio of completion percentage gained per dollar spent does not imply that the option is also feasible; the total cost to achieve such a ratio may exceed the allocated budget.

Figure 10 shows the total savings (and in the presence of negative values, total losses) of grid solutions relative to the costs of the adding processors to the primary servers. However, this figure does not account for a grid's initial cost, as there are no such prices presently available. The most prominent savings occurred among the Itanium 2, POWER4+ and Sun UltraSparc processor families. Given the large savings among these families, either Pentium III or Pentium 4 grid solutions would be financially intelligent investments for meeting customer demands during peak times. Within the Xeon families, the Pentium 4 grid computing solutions showed to be more expensive than upgrading adding a second processor to the primary server. Although Pentium III grid solutions were cheaper, after accounting for the effort put into configuring a grid, it is probably neither a financially nor a computationally wise investment.

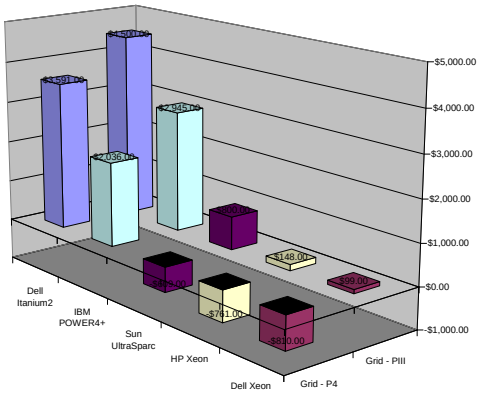


Figure 10. Total financial savings of grid computing relative to upgrading the primary server.

While there are certainly some remaining open sections in terms of cost analysis, such as the cost of grid software itself, there is not much room for further analysis because such prices are not available. For small and medium e-Businesses, we assumed that the initial support and maintenance costs would be comparable for grid and non-grid solutions, and thus this study ignored these costs. We also assumed that companies could utilize current network infrastructures if a grid solution was realized, and therefore omitted additional costs related to network topology.

4.5 Scalability, Cost and Performance

One of a grid's key features is its scalability. We showed that for the selected common small and medium e-Business resources, grid computing solutions were competitive performance-wise (via Figure 6) with larger servers when employed correctly (i.e., given the available network technology, the replicated data sizes fall within acceptable limits). But even more importantly, single servers cannot be upgraded indefinitely. During the lifetime of a growing e-Business, there will eventually come a time when the primary server's processing capabilities will be exceeded. Grid computing solutions work to avoid this bottleneck by allowing e-Businesses to incrementally grow the amount of available processing power. This is done by adding secondary nodes over a local or wide area network. Figure 11 shows the total costs of grid solutions versus a non-grid solution when using Dell Xeon

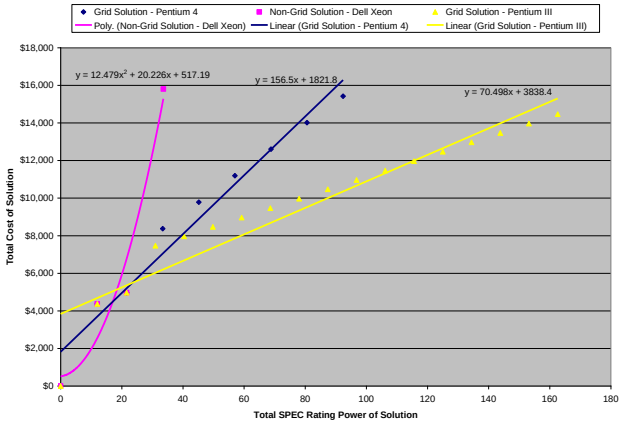


Figure 11. Total cost of various e-Business solutions.

server technology (note that because no actual costs are available, it was assumed that the initial costs of grid solutions were $2,000). For each solution, the raw data (queried from Simulation 3, Simulation 4 and Simulation 5 via the Appendix) is portrayed with a scatter plot and accompanied by a trend line. It is shown that the cost of grid solutions grow at a much more linear rate and can provide more compute power than a non-grid solution. Using Figure 6 and Figure 11 as evidence, it is apparent that grid solutions can provide more processing power at a cheaper cost and yield similar levels of throughput than their non-grid counterparts. In the case of Dell Xeon technology, after the first $5,000 spent, grid solutions become much more beneficial than non-grid solutions in terms of available processing power per dollar spent. While the grid definitely scales well for the provided examples, we cannot make further claims regarding the price and scalability without further studies. Grid computing also has the added benefit that these additional resources can be located in different geographical locations. This is especially useful for companies looking to expand, all while making efficient use of available resources, uniting to appear as one large virtual organization.

Geographical independence leads to a topic not addressed by this research. It permits the possibility of leasing resources from other companies during peak times. Rather than an e-Business instantiating their own grid system, processing power can be leased from other companies that have processing power to lend. This has the potential to be a very attractive option, an option that is not available to non-grid solutions. Depending on the cost per unit processor leased, it may be easier and less expensive for an e-Business to lease their additional processor power during their peak usages, but this still requires that the client applications implement an open-standard grid interface in order to coordinate with grid service providers.

5. CONCLUSIONS

Several quantitative measurements have been made which show that grid computing, if correctly employed, can offer small and medium e-Businesses financial benefits over upgrading the primary server. We showed that grid computing is most beneficial to companies whose primary servers' processors are Itanium 2 and POWER4+, offering total savings up to $4,500 (via Figure 10) in the case of the fictitious e-Business. As a first step, grid computing solutions did not financially benefit the Xeon processor configurations. However, it would certainly provide scalable solutions for future upgrades that reach beyond the addition of a second processor. We also showed that companies need to determine how much information will potentially be shared over a grid. In the case of the online housing search e-Business, once solutions breached 200 KB/subtask, cost and throughput fell far short of acceptable limits, making grid solutions inappropriate for that particular network infrastructure. While not all solutions will become infeasible at this transfer size (e.g., different sites have different networking capabilities), it defines a focal point for attention.

While these experiments ignored the cost of maintenance, this is an issue that definitely requires further examination in a future study. Other future work includes re-evaluating the experiments when the SPEC CPU (INT) 2004 results are publicly available with newer technology. We would also like to work with a real-life company whose environment is sufficient for the use of grid computing. We also look to setup a small grid using the Globus Toolkit framework to examine first-hand results of a grid and how it affects the company's profits and throughput during peak times.

6. ACKNOWLEDGMENTS

First and foremost, I would like to give my extended thanks to all of my peers and professors that have participated in reviewing this paper through its several drafts over the past several months. I would also like to give a very special thanks to my advisor, Ann Smith, who has been very helpful with all of her thoughts and suggestions in guiding this research to make it its finest. Last but not least, I would like to acknowledge the members of the GridSim team, Dr. Rajkumar Buyya and Anthony Sulistio, at the University of Melbourne for not only creating a wonderful grid simulator application, but also for being very helpful by answering questions regarding the simulator.

7. REFERENCES

[1] Schaller B. The Origin, Nature, and Implications of Moore's Law [online] 1996. Available from: http://mason.gmu.edu/~rschalle/moorelaw.html. Accessed 2004 Mar 5.

[2] Haynos M. Perspectives on grid: Grid computing – next-generation distributed computing [online] 2004. Available from: http://www-106.ibm.com/developerworks/grid/library/gr-heritage/. Accessed 2004 Mar 3.

[3] Scannell E. Coming to Grips with Grids. *InfoWorld* 2004 Jan; 42.

[4] Foster, I. What is a Grid? A Three Point Checklist. GRIDToday, July 20, 2002.

[5] Buyya, R., Murshed, M. GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing. The Journal of Concurrency and Computation: Practice and Experience, Volume 14, Issue 13-15, Wiley Press, Nov.-Dec., 2002.

[6] Weissman J. Grids in the Classroom [online] 2000. Available from: http://dsonline.computer.org/archives/ds300/ds3eduprint.htm Accessed 2004 Mar 9.

[7] Buyya, R., and Murshed, M. Using the GridSim Toolkit for Enabling Grid Computing Education, International Conference on Communication Networks and Distributed Systems Modeling and Simulation (CNDS 2002), January 27-31, 2002, San Antonio, Texas, USA.

[8] Foster, I., Bester, J., Kesselman, K., et al. GASS: A Data Movement and Access Service for Wide Area Computing Systems. Sixth Workshop on I/O in Parallel and Distributed Systems, May 5, 1999.

[9] Stockinger, H., Samar, A., Allcock, B., et al. File and Object Replication in Data Grids. Journal of Cluster Computing, 5(3)305-314, 2002.

[10] [Anonymous]. SPEC CPU2000 Press Release FAQ [online] 2004. Available from: http://www.spec.org/cpu2000/press/faq.html. Accessed 2004 Feb 28.

8. APPENDICES

8.1 Simulation Descriptions

8.1.1 Simulation 1

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	Dell PowerEdge 3250	Itanium 2 1.5 GHz	2	$12,999	1	12.7

8.1.2 Simulation 2

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	Dell PowerEdge 3250	Itanium 2 1.5 GHz	2	$17,999	2	25.4

8.1.3 Simulation 3

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	Dell PowerEdge 1750	Xeon 3.06 GHz	1	$4,367	1	12.0

8.1.4 Simulation 4

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	Dell PowerEdge 1750	Xeon 3.06 GHz	2	$4,966	2	21.6

8.1.5 Simulation 5

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	Dell PowerEdge 6650	Xeon 3.06 GHz	4	$10,999	4	33.9

8.1.6 Simulation 6

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	HP ProLiant BL20p G2	Xeon 3.06 GHz	2	$5,750	1	14.0

8.1.7 Simulation 7

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	HP ProLiant BL20p G2	Xeon 3.06 GHz	2	$6,398	2	25.8

8.1.8 Simulation 8

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	IBM pSeries 615 Model 6C3	POWER4+ 1.2 GHz	2	$6,540	1	8.1

8.1.9 Simulation 9

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	IBM pSeries 615 Model 6C3	POWER4+ 1.2 GHz	2	$9,985	2	15.2

8.1.10 Simulation 10

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	Sun Blade 2500	UltraSparc 1.28 GHz	2	$7,595	1	7.01

8.1.11 Simulation 11

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	Sun Blade 2500	UltraSparc 1.28 GHz	2	$8,395	2	12.4

8.1.12 Simulation 12

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	Dell PowerEdge 3250	Itanium 2 1.5 GHz	2	$12,999	1	12.7

8.1.13 Simulation 13

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	Dell PowerEdge 3250	Itanium 2 1.5 GHz	2	$12,999	1	12.7
2	Dell Precision Workstation 420	Pentium 3 1.0 GHz	2	$499	2	9.40

8.1.14 Simulation 14

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	Dell PowerEdge 1750	Xeon 3.06 GHz	2	$4,367	1	12.0
2	Dell Precision Workstation 340	Pentium 4 2.80 GHz	1	$1,409	1	11.8

8.1.15 Simulation 15

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	Dell PowerEdge 3250	Xeon 3.06 GHz	2	$4,367	1	12.0
2	Dell Precision Workstation 420	Pentium 3 1.0 GHz	2	$499	2	9.40

8.1.16 Simulation 16

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	HP ProLiant BL20p G2	Xeon 3.06 GHz	2	$5,750	1	14.0
2	Dell Precision Workstation 340	Pentium 4 2.80 GHz	1	$1,409	1	11.8

8.1.17 Simulation 17

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	HP ProLiant BL20p G2	Xeon 3.06 GHz	2	$5,750	1	14.0
2	Dell Precision Workstation 420	Pentium 3 1.0 GHz	2	$499	2	9.40

8.1.18 Simulation 18

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	IBM pSeries 615 Model 6C3	POWER4+ 1.2 GHz	2	$6,540	1	8.1
2	Dell Precision Workstation 340	Pentium 4 2.80 GHz	1	$1,409	1	11.8

8.1.19 Simulation 19

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	IBM pSeries 615 Model 6C3	POWER4+ 1.2 GHz	2	$6,540	1	8.1
2	Dell Precision Workstation 420	Pentium 3 1.0 GHz	2	$499	2	9.40

8.1.20 Simulation 20

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	Sun Blade 2500	UltraSparc 1.28 GHz	2	$7,595	1	7.01
2	Dell Precision Workstation 340	Pentium 4 2.80 GHz	1	$1,409	1	11.8

8.1.21 Simulation 21

Resource Identifier	Machine	Processor type	Maximum processors	Cost	Number processors used	SPEC rating
1	Sun Blade 2500	UltraSparc 1.28 GHz	2	$7,595	1	7.01
2	Dell Precision Workstation 420	Pentium 3 1.0 GHz	2	$499	2	9.40

Enhancing Digital Rights Management using the Family Domain

Michael Brogan
Saint Mary's University
700 Terrace Heights
Winona, MN 55987
mfbrog00@smumn.edu

ABSTRACT
Digital Rights Management is a system for managing the distribution and control of copyrighted digital data. Many new approaches to implementing Digital Rights Management attempt to address the problems wrought by the rampant peer-to-peer sharing of digital music files facilitated by programs such as the original Napster. However, current and legal online distribution systems, such as iTunes and MusicNet, use proprietary Digital Rights Management solutions that lack scalability and impinge on the privacy and freedom of users. Our research proposes a specific implementation for Digital Rights Management to address these problems, which is based on the recently proposed general concept of the Family Domain. Initial tests demonstrate that our implementation has good potential to address issues associated with current Digital Rights Management implementations while still providing adequate management and control.

Categories and Subject Descriptors
K.6.5 [**Management of Computing and Information Systems**]: Authentication. K.4.1 [**Public Policy Issues**]: Intellectual Property Rights, Privacy. K.4.4 [**Electronic Commerce**]: Distributed commercial transactions.

General Terms
Design, Security, Standardization, Legal Aspects.

Keywords
Digital Rights Management, cryptography, security, digital content, copyright protection, mobile device.

1. INTRODUCTION
Digital Rights Management (DRM) refers to the process of a content provider managing the "rights" of consumers (end users) to distribute and render digital files. This includes "the technologies, tools, and processes that protect intellectual property during digital content commerce" [1]. DRM is an abstract solution with many current implementations arising from the need to protect digital content while ensuring (and maintaining) its authenticity (legal and otherwise). DRM technologies are being used to handle "intellectual property" [2] concerns such as piracy, security, and the ability to offer potential benefits to consumers (such as specialized content) [4].

Proceedings of the 4th Winona Computer Science Undergraduate Research Seminar, April 20–21, 2004, Winona, MN, US, Copyright 2004.

The emergence of peer-to-peer file sharing programs and unmonitored/unmediated file distribution networks (including programs such as KaZaa, the original Napster, and Gnutella) has lead to rampant piracy of digital music and lost profits for record companies and the artists that they represent [6]. In an attempt to combat these issues, legal online music distribution systems (such as iTunes, MusicNet and the new Napster 2.0) use new, proprietary DRM implementations.

These new DRM implementations provide a good starting point offering the control and functionality required by providers of copyrighted digital content [5]. Content providers will continue to evolve. They will increase quantity of items in their content libraries, enhance the software used to distribute the content, and improve the content's security. The consumer, however, is still left wanting. Current DRM solutions, from the consumer viewpoint, are not very scalable and offer little in the way of freedom and privacy of use. End users want to be able to use music and other digital content in an unrestricted and unmonitored manner after they have purchased the rights to such content [3].

For example, current DRM systems severely restrict the number of computers and external devices on which end users can manage and render their files. These restrictions are clearly contradictory to section 107 of the U.S. copyright law known as the Fair Use Doctrine. Under the Fair Use Doctrine, consumers have the right to use their purchased content as they see fit, as long as such use does not severely interfere with the person(s) that hold the right(s) of that intellectual property. In other words, a consumer of digital music who has purchased the rights to that music can use that music as he or she sees fit, as long as it does not severely interfere with the musician who holds the copyright [11]. Because the definition of "severely interfere" is open for liberal interpretation, fair use has been used as rationale for the distribution of music files over peer-to-peer networks. As the music industry lashed out against these networks, the unfortunate result was that DRM implementations were extremely restrictive and they have not adequately taken the idea of fair use in account.

Our research centers around the Family Domain (FD) concept, which was introduced as a potential strategy for providing more flexible consumer control over distributed digital content [8]. The FD concept has yet to be implemented as part of a complete DRM solution and only parts of the FD concept have been implemented and integrated into systems that are currently entering the market [10]. This research will test the assumption that a particular and complete implementation of the FD concept can, not only effectively address the issue of digital rights, but can also provide scalability of use and flexibility of control.

This paper is organized as follows: Section 2 provides a discussion of background research necessary to understand current state of DRM, in general, and the FD concept, in particular. Section 3 shows proof of concept by specifying our

design for a FD implementation using current technologies. Section 4 explains the importance of such an implementation and the methodology of this paper. Section 5 compares our proposed FD implementation to currently existing DRM implementations in terms of the enforcement of digital rights and consumer usage parameters. We end by offering a perspective on where our implementation of the FD fits into the overall scheme of digital rights protection and suggestions for future work.

2. BACKGROUND RESEARCH: ORIGINAL FAMILY DOMAIN

In order for DRM technologies to survive, an acceptable strategy must be found that all parties (rights holders, content providers, and consumers) can agree provides adequate security while respecting the idea of consumer fair use [7]. The FD abstraction offers a general approach to incorporating the idea of fair use in DRM systems.

This original FD concept was introduced by Motorola as a submission to the Open Mobile Alliance (OMA) for use with mobile phones [8]. There is currently a draft of the OMA's forthcoming implementation, Specification 2.0, which has incorporated portions of the FD approach. This specification is only in a drafted form and simply outlines that a FD of devices can be created for use with content from mobile phones. As of yet, this specification does not provide any new insight to add to the FD concept. It does, however, suggest that the topic is important and that companies will begin to incorporate portions of the FD into DRM implementations.

Most current DRM implementations are such that they require a centralized locker approach to give users access to their content. However, many of these previous approaches are not suitable for devices that lack permanent networking capabilities. The FD offsets the centralization that currently exists by offloading it to a Domain Authority (DA). The DA is a "server" that installs a common DRM private key on each of the user's devices. The DA would be the device that is central to the FD. It is through this device that DRM transactions would be monitored and controlled. In essence, the current concept of the DRM private key (unique digital keys that would be associated with each user device to identify the device as authentic and able to use protected content, one key for each device) becomes a domain private key that enables access to all the content within a domain (a single, yet unique, digital key that would be used by all of a user's devices). A secure perimeter is established and devices inside the domain have full access to the content associated with the key. See Figure 1 for an illustrated representation of this. In the figure, the arrows represent a single key obtained from a content provider and passed to each of a user's devices within a FD and not a friend's PDA. This construction is thus suitable for devices without permanent networking capabilities. Any device (such as a set-top box, cell phone, PDA, or MP3 player) would only have to register with a DA once. These devices would not need to obtain a key from the content provider, only the DA (the "server" that originally obtained the single key from the content provider). Instead of dealing with content providers on a play-by-play basis for each device, users would only have to ensure the initial membership of their devices within the FD (membership granted by the DA). [8]

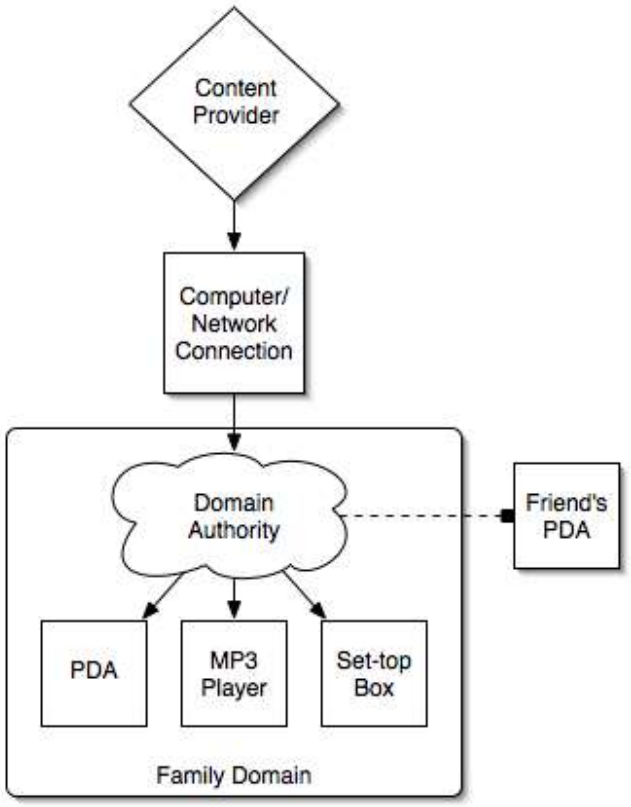


Figure 1 - Family Domain Key Passing

This paper expands upon the initial FD abstraction and provides the details for implementing a DRM system that uses this expanded abstraction.

3. FURTHERING RESEARCH: EXPANDED FAMILY DOMAIN

The FD is to be a trusted system and all devices in the system would be allowed to use protected content. Although this general outline is a great step forward toward a more adequate DRM technique, it leaves out concerns for user privacy. This is where the proposed discussion of the FD in this paper begins to diverge from, and expand upon, the general outline that was presented in the Motorola submission. The usage habits of an individual user could be monitored very easily in an implementation such as original family domain. The divergence proposed here is one that puts more trust in users and does not relay the usage characteristics within the FD back to the content providers. Both strategies, the original FD approach and this new Expanded Family Domain (EFD) approach, are designed to allow for more scalable usability of content but the EFD also promotes more privacy for users.

The EFD approach suggests that the only tie to a content provider would be the unique key associated with a user's account. This simplicity of the EFD approach would also impact the DA introduced in the original proposal by simplifying its duties. Both approaches will plainly give more freedom to end users by allowing them to be in control of what devices they have associated with their DRM account through an intermediary, the DA. However, the EFD takes this control even further. It simplifies matters by not requiring a unique hardware ID to be

obtained from a device by the DA. The EFD approach can be thought of as a waterfall in terms of device authentication: A key flows down to a device and the device does not need to send any information back up stream.

We are now at a point where the strengths of the EFD can be more clearly contrasted against the issues of current DRM implementations and even the original FD proposal. The EFD approach directly affects the main problems associated with these types of services and strategies:

- It reduces the complexity of authentication. It suggests one time authentication between the DA and the content provider giving users their unique account key. This is done instead of authenticating with a content provider every time a device is added to, or removed from, an account. It is further simplified by not requiring a unique hardware ID to be obtained from a user's device.
- It reduces the ties of user information to content servers. Only the unique account key is associated with content servers. This is done instead of listing every device owned by a user on content provider servers (effectively dealing with the issues of possible marketing exploitation and privacy [3]).

And the EFD maintains the following goals from the original FD proposal:

- It reduces the complexity of obtaining content. Content purchased on any device within the domain will work on any device within the domain. This eliminates the need for every device to obtain authentication from content providers.
- It allows users fair use of content while still requiring the content to be associated with an account. It provides checks to ensure that such content will only be rendered on account-authorized devices.

In order to more clearly delineate between FD and the EFD, the following sections will provide the design and workings of our approach.

3.1 Domain Authority (DA)

It is assumed that the EFD and all associated components can be implemented using current technology. Hardware devices that exist today can be used with this distribution model by modifying their software/firmware to allow for key acquisition (obtaining the unique key required to render content within the realm of the EFD). The DA can be implemented in the form of software running on any computer with the appropriate connections (USB, FireWire, Bluetooth, etc.) which portable media devices use to connect. The DA software will act as the mediator for all devices within the domain. This model (instead of a hardware implementation of the DA or some other incarnation) would allow for a more rapid introduction of the concept and minimize the associated costs by not requiring any new hardware research or development. This model is pictured in Figure 2. Note the incorporation of the computer into the concept of the DA as part of the EFD concept introduced in this paper. The original FD paper left the DA as an abstracted concept.

The purpose of the DA remains the same: It will manage the content within an EFD. This concept is similar to a traffic cop who works for a police department but manages a designated traffic intersection (the police being the content provider, while the intersection is the area within the EFD). The DA is a standalone entity that uses the rules from its employer (the content provider) to safely and orderly manage the traffic (devices in the EFD).

From start to finish the DA provides the following functionality: When a user first installs the DA software on the machine, the software allows for the creation of an account. The DA creates this account with a content provider and obtains a unique account key for use within the EFD. This process involves obtaining the rules that the DA will use to manage devices within the domain (amount of devices existing in the domain, amount of time between additions and removals of devices, or restrictions on which types of devices can be added). This process then gives the DA software the authority to manage devices that abide by the set rules. Thus ends the communication between the DA and the content providers. Now, whenever the computer or any device obtains a secured piece of content, it is encoded with the account key and will only play on devices within the specified domain. Although the DA is a rule-maker, as in the original FD design, the EFD focuses this by making the DA a piece of software.

Due to the nature of the DA, the software must be highly trusted. It must be modeled (implementation details aside) in such a way that malicious individuals would have a minimal impact on the security of the key and it must hinder the ability to spoof devices. Thus, the software must be easily updateable and users may periodically be required to update their management software if they wish to further add or remove devices.

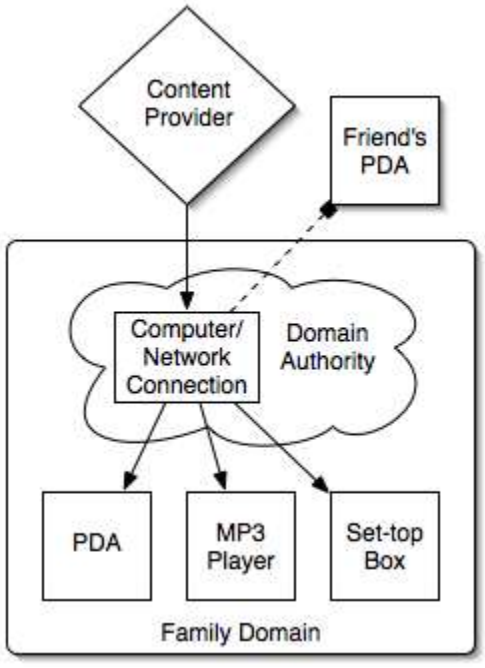


Figure 2 - Expanded Family Domain

3.2 User Devices

User devices consist of any device on which software can be installed or modified to allow for the distribution model. This includes cell phones (as introduced in the former FD model), set top boxes (such as stereos or home theatre components), Personal

Digital Assistants, MP3 players (which in fact would play MP3s *and* the encoded content which is distributed via the DA model), and car stereos. In order for these devices to function, they must contain the software that checks their EFD key against the key encoded with the digital content.

This software could take various forms. For the EFD implementation, the suggestion is minimalistic. For devices with an operating system, the software will simply be an addition that manages I/O of only the encrypted content.

3.3 Content Providers

Work done by content providers using an EFD approach would also be simplified (in comparison to current DRM or even the original FD approach). The database for user accounts would be smaller and only contain preset rules. Current implementations, such as Apple's iTunes, contain a database of all hardware associated with an account, the associated keys, and other information identifying users and their payment information. Focusing on privacy, the EFD approach would simply contain the account key with the user information and payment information.

The original FD approach varies greatly from the EFD in this respect. It does not explicitly remove these connections of user information that have the potential for commercial exploitation [3]. This was not a concern in the initial conceptualization.

3.4 Other Features

The EFD distribution model also has potential for additional usage benefits to be easily implemented. One such feature is content tagging. This is the ability to listen to a song on a radio (car or set top box) and tag the content for later (or immediate) download. In this model, the content would only be encoded with the account key associated with the tag and could then be rendered on any device in the user's EFD [8].

4. METHODS

4.1 Importance of the Family Domain

It is important to show the effectiveness of a revision of DRM strategies and discuss how the EFD solves the problems inherent in the current schemes. In other words, this paper provides a DRM strategy where users', distributors', and copyright holders' desires coincide. The implications of this paper, and how this new structure is beneficial, are far reaching. The concept of the EFD can be extended to other digital content types, such as textual documentation (electronic books) and even software.

Illustration of the improvements has already begun to be accomplished as the design of the EFD and DA has been discussed and their benefits shown. The methods for showing the effectiveness of this solution, so far, have relied upon faith that the aforementioned discussion is reliable and appropriate to the DRM domain. The arguments presented also assume that this theoretical implementation of the EFD can be implemented as it has been outlined. Nonetheless, the EFD has plainly begun to show its advantages in terms of scalability and ease of implementation.

The methodology of this paper has been to introduce the FD concept in Section 2 as an important and innovative concept and then more clearly define the FD in terms of the EFD in Section 3. This paper borrowed only the general structure of the FD as outlined in the original paper by Motorola. The detailed design of the DA provided in Section 3 has begun to show how the FD concept is important and that it can be more readily shown as feasible in this discussion of the EFD. Thus, elucidation provided by the EFD has, so far, met the goal of this paper: To show that the EFD not only addresses the issues associated with digital rights, but also provides more scalability and flexibility than current implementations.

However, to truly show the advantages of the EFD DRM technique, concrete comparisons to current schemes are also needed to supplement this discussion. These comparisons will use the EFD approach and will be the "experimental" portion of the paper.

4.2 Comparing the Family Domain to Other Digital Rights Management Techniques

The advantages of the EFD concept have been shown as follows:

- Outline structure and expand on research already done on the FD (done)
- Explain obvious advantages of a good implementation of the EFD (done)
- Detail the design and workings of the EFD DRM approach (done)

The advantages of the EFD concept have yet to be shown in the following way:

- Compare security and usability of the EFD to other implementations

Due to a lack of funding and time (and the fact that the concept is still in a theoretical stage), this paper is only an ordering of the concepts. However, this proof-of-concept paper concretely shows the benefits of the EFD in terms of scalability, usability, and lowered complexity for end users while still providing adequate security measures for content providers.

To summarize, the strategy has been to introduce the FD and more clearly show its importance by expanding the concept and following it up with the EFD. The EFD concept has been an incarnation unique to this paper, though it borrows the general structure of the original FD: A domain in which a family's devices will all render legally acquired content. All arguments contained herein have been to further show the importance of the concept and why the EFD would be more acceptable. The goal of this paper is not to replace the FD but, instead, to expand upon a good concept that was only briefly discussed in order to give it more relevance in the realm of DRM and consumers' concerns.

5. RESULTS AND ANALYSIS

Security measures equal to those of iTunes, the OMA 2.0 specification, or any other DRM technique can be easily implemented using the EFD concept. However, before such security measures are discussed, a further analysis of the differences between the EFD DRM distribution approach and current implementations will be revealed.

The following section contains the arguments and experimentation done to show, concretely, that the EFD approach is an adequate form of DRM that also provides scalability of use and flexibility of control unmatched in current DRM implementations.

5.1 Content Acquisition

A limiting factor of current DRM implementations is the type of content acquisition that is used. As an example of a real-world implementation, iTunes will be used. This is, unarguably, the least restrictive and provides the best candidate for experimental comparison. The iTunes approach to content acquisition requires a host computer (Macintosh or PC) capable of running the iTunes application that facilitates the download of protected content from the Apple content server. Content acquisition is currently limited in this way with all DRM implementations because their structure associates with the computer first and foremost as an intermediary (though wholly different than the DA). As seen in Figure 3, the EFD approach would significantly improve the ability to acquire content, as each device within the domain would already contain the required key to render protected content. If a network connection exists for any device within the domain, it would have the ability to acquire content. Likewise, content can easily be transferred from device to device if a medium for such communication exists.

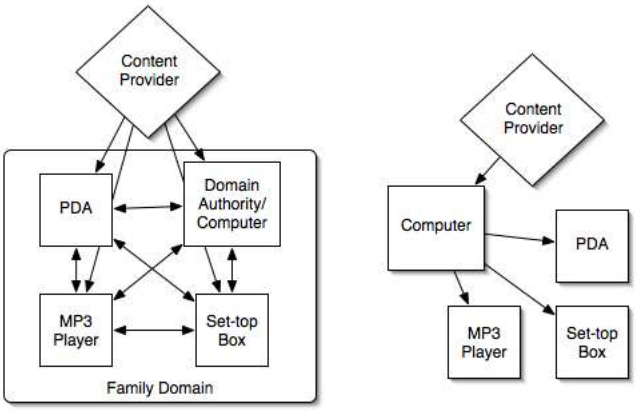


Figure 3 – Expanded Family Domain vs. Current DRM Acquisition

For instance, imagine that a PDA and MP3 player both have a CompactFlash card slot and a consumer has an EFD protected file on her CompactFlash card. The CompactFlash card can be moved between devices without needing to communicate with the user's computer (much like the unrestricted DRM-less MP3 format). This exemplifies a DRM technique that works transparent to the end user. All of a user's devices can easily transfer files, but if the same CompactFlash card were used in a friend's PDA, the file would not play, because the friend does not have the account key on his PDA.

This concept, although simple, would be an extremely important and significant move towards a strategy that allows an end user the freedom they deserve (in terms of fair use) while still providing protection from mass piracy. Like the example of the friend's PDA, the files associated with a user's account would be useless if shared over a peer-to-peer network such as KaZaa. Any users who download the file would have no means to render the protected content. These results show that the EFD concept would simply be more adequate at facilitating acquisition by multiple devices and sharing the acquired content within the domain. Current implementations provide an approach that is too complicated and highly restrictive requiring the use of a computer every step of the way.

5.2 Information Flow

In a world where vast amounts of, potentially exploitable, user information are available, privacy is very important for end users [3]. The current DRM implementations allow these vast amounts of usage statistics to be uploaded to content providers. This includes what files are used, how often they are played, and on what devices. It should also be noted that iTunes is one of the least intrusive services. Many others provide abhorrent amounts of usage information to content providers [9].

To more clearly show the results of this experiment, Figure 4 provides a diagram showing how the EFD approach to DRM would significantly limit the amount of usage information available to content providers. The arrows in the diagram indicate the direction of the flow of information. In the EFD approach, information does not flow beyond the user's domain, usage is not tracked, and the user is trusted. This is one of the main features of the EFD and differs greatly from the original FD concept. However, as part of the implementation of the iTunes service, information must constantly flow back to the content provider in order to associate devices with a user's account. Also, please note that iTunes only facilitates sharing between a Computer and MP3 player, specifically the iPod. Thus, the images used here are generous and one would hope that current implementations would expand to incorporate other devices in the near future.

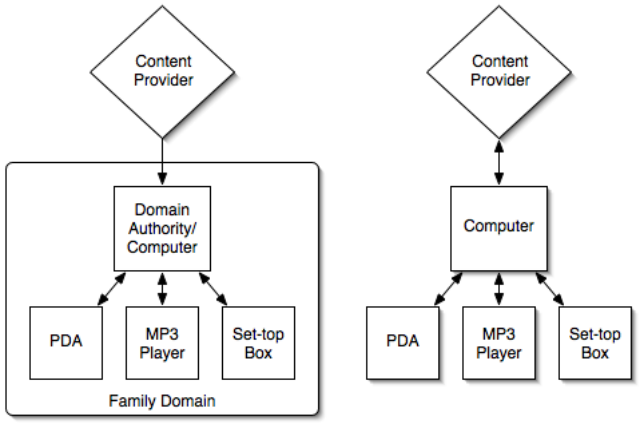


Figure 4 – Expanded Family Domain vs. Current DRM Information Flow

The EFD approach again shows its strength as privacy concerns are taken into consideration. However, one must concede this concept would be a radical shift for content providers to adopt, as the trend has been to gather as much information about users as possible. Therefore, the EFD approach is optimistic that corporations will eventually adopt approaches with consumers' concerns in mind. The results of this experimental comparison show how information flow would more adequately facilitate the goal of privacy for end users in the EFD approach compared to current implementations and even the original FD proposal.

5.3 Usability Concerns

In addition to the aforementioned differences between the current DRM techniques and the EFD, there are other usability concerns that must be taken into account. The software available for current DRM strategies facilitates acquisition of content as previously discussed. Table 1 shows the limitations these software programs and DRM techniques place on downloaded

content. In terms of usability, a consumer would most likely desire a solution that mimics the usability of a common CD over a more restrictive method [9]. This table is a modified version comparing previous research done on current DRM implementations. It has been changed to clearly show how the EFD approach would more adequately meet the strengths that are associated with a compact disc.

	# of CD burns per purchase	# of portable device transfers per purchase	# of computers per purchase	offline access to user purchased tracks	relationship required for playback
iTunes	∞	∞	3	FALSE	TRUE
PressPlay	1	1	2	FALSE	TRUE
Rhapsody	1	0	∞	TRUE	TRUE
MusicNet	2	0	2	TRUE	TRUE
MusicNow	2	2	3	FALSE	TRUE
Liquid Audio	3	3	1	FALSE	TRUE
CD	∞	∞	∞	TRUE	FALSE
Family Domain	∞	∞	∞	TRUE	FALSE

Table 1 – Permissions of DRM Techniques (modified) [9]

As the table clearly shows, the EFD approach is just as limited as a CD when considered in the domain of a user. Yet, the EFD provides copy protection techniques that a CD does not have in an attempt to appease both the corporations and real people in the world. The EFD is truly the more acceptable approach. The results shown in the column labeled "relationship required for playback" are some of the most significant. Because the EFD account is managed within a user's own environment (on their computer), a user may cease communication with a content provider and forever have the ability to use their purchased content.

Therefore, combined with the EFD approach introduced in this paper, this table clearly shows the usability advantages of a more scalable structure. The paper from which the chart was taken maintained that a DRM approach with the usability of a CD might finally provide an acceptable DRM solution [9]. This scalability of use is exactly what the EFD has intended to facilitate.

5.4 Security Concerns

Security for content providers is also an important part of the equation. The entire goal of DRM techniques is to restrict content from being illegally distributed in mass quantities and the EFD approach, as a DRM technique, must comply. This would be met in terms of incorporating encryption into the files so that the account key can never be read, manipulated, or copied. It would need to be no different than the encryption incorporated in technologies such as the protected AAC files used by iTunes or the protected WMA files used by Windows Media based services. Therefore, no further work would need to be done to ensure this level of compatibility and reliability. In essence, even if the encryption technique were identical to one of the current methodologies, the EFD approach will provide a vastly superior distribution technique. The encryption techniques available today, although important, are at a level acceptable to consumers and corporations alike. Associating the EFD with any current encryption technology would require the following data to be encrypted: A unique user account number, metadata about the account, and a watermarking system to ensure that content is authentic.

The important factor to note is that security concerns have already been dealt with. The contribution of the EFD is the approach to distribution and management of content and protection will still be in place on a per file basis. As long as the application written for the EFD remains trusted, the content will meet or exceed any current DRM security implementation. More specifically, the EFD concept can be applied using the encryption techniques employed by current implementations providing an even quicker route for implementation.

As a side note, passwords could be optional with such a strategy because the domain itself provides the trust needed. Passwords could easily be implemented and used when acquiring content from a provider. Again, only the unique account key will be required to render content, the password would merely be used to limit the ability to obtain content and would be used when purchasing from a content provider. However, their removal (or optional use) from the scheme would allow trusted devices without keyboards to more easily obtain content from providers.

6. CONCLUSIONS

This research has shown that the EFD distribution model is superior to the current standing approaches in terms of its low complexity for the end user, scalability in terms of use, flexibility of control, and adequate security in its ability to theoretically perform as well as the current technologies. The results of the experiments have clearly shown how content acquisition, information flow and usability demonstrate the EFD's strong potential as a replacement for the current strategies.

The results of this research show that the EFD approach offers a solution that will adequately meet the needs of both end users and content providers. This concept and its introduction by the employees of Motorola could mark a shift in the societal understanding of digital downloads. If end users are sufficiently satisfied with their ability to use the content as they see fit, it can be surmised that they would be willing to pay for their content. Popular media criticizes the current content providers for the very reasons that the FD was founded to resolve.

6.1 Further Research

This paper does not rely on a simulation of the EFD, or any form of its implementation, but infers that such continuation would be worthwhile and beneficial. Thus, the suggested path to follow up on this paper would be to prototype the EFD and associated DA to further prove its applicability and potential as a solution for a future DRM implementation.

7. ACKNOWLEDGMENTS

This paper was accomplished with the background research and further help from Thomas Messerges and Ezzat Dabbish of Motorola Labs as well as guidance from Ann Smith of Saint Mary's University. Drafts of the upcoming OMA were also very beneficial in completing this research and were provided by Balazs Kiacz Forapolis, an OMA technical paper staff member.

8. REFERENCES

[1] Association of American Publishers, A. *Digital Rights Management for Ebooks: Publisher Requirements*, 2000.

[2] Baase, S. *A Gift of Fire*. Prentice Hall, 2002.

[3] Cohen, J. DRM and Privacy. *Communications of the ACM*, Vol. 46, No. 4, April 2003.

[4] Fetscherin, M. Schmid, M. Comparing the Usage of Digital Rights Management Systems in the Music, Film, and Print Industry. *ACM* 2003.

[5] Garnett, N. Digital Rights Management, Copyright, and Napster. *ACM* 2003.

[6] Kwok, S. Digital Rights Management for the Online Music Business. *ACM SIGecom Exchanges*, Vol. 3, No. 3, August 2002.

[7] Liu, Q., Safavai-Naini, R., Sheppard, N.P. Digital Rights Management for Content Distribution. *Conferences in Research and Practice in Information Technology*, Vol. 21., 2002.

[8] Messerges, T., Dabbish, E. Digital Rights Management in a 3G Mobile Phone and Beyond. *DRM'03*, October 27, 2003.

[9] Mulligan, D. Han, J. Burnstein, A. How DRM-Based Content Delivery Systems Disrupt Expectations of "Personal Use". *DRM'03*, October 27, 2003.

[10] Open Mobile Alliance Ltd. *Technical Specifications*, 2004.

[11] Samuelson, P. DRM {And, Or, Vs.} The Law. *ACM* 2003.

Home Wireless Networks: Performance under Interference

Andrew Schaff
Computer Science Department
Winona State University
Winona, MN 55987
apschaff3432@webmail.winona.edu

ABSTRACT
As technologies advance in the field of communications, more users are turning to wireless. Whether it's cell phones, personal data assistants (PDA's), or connectivity to the Internet, wireless communication has become a popular aspect of society. More specifically, home Internet users are configuring wireless networks (WI-FI). Like all other wireless technologies, WI-FI broadcasts its signal at a certain frequency, leaving itself susceptible to other devices broadcasting at similar frequencies. This paper focuses on testing the throughput, transaction rate, and response time of an 802.11b home WI-FI network with interference administered to it. A list of devices broadcasting at similar WI-FI frequencies were used to create the interference subjected to the network. Only under heavy interference using multiple broadcasting devices did the wireless network falter in performance. Analysis of our results concludes that 802.11b wireless networks in a home environment have a 17% to 30% drop in performance under moderate interference conditions.

Categories and Subject Descriptors
C.2.3 [**Network Operations**]: Language Constructs and Features – *network management, network monitoring, public networks*

General Terms
Measurement, Performance, Experimentation.

Keywords
WI-FI, WLANs, IEEE 802.11b, interference, frequency, throughput, response time, transaction rate, broadcasting devices, wireless performance

1. INTRODUCTION
Wireless communication is becoming a popular commodity in the home setting of the 21st century. In fact, a 2001 study by Allied Business Intelligence stated a $120 million dollar increase in wireless business from 2000. "By 2006, the global market is expected to be worth $2.4 billion" as well as wireless accounting for 48 percent of nodes [1]. More specifically, home Internet users are configuring wireless networks (WI-FI) to meet their needs. These needs range from reducing complications of setup to freedom of mobility to a more simplistic, wire-free approach. The Institute of Electrical and Electronics Engineers' (IEEE) implementation of the 802.11 standard for WI-FI was the first step. 802.11 specifies an over-the-air interface between a wireless client and a base station or between two wireless clients [1]. Like all other wireless technologies, WI-FI broadcasts its signal at a certain frequency, leaving itself susceptible to other devices broadcasting at similar frequencies. Our scope focuses specifically on the popular home standard, 802.11b, which broadcasts at 2.4 GHz.

This paper is geared towards testing the throughput, transaction rate, and response time of a home WI-FI network with interference administered to it. Devices which broadcast at similar WI-FI frequencies created the interference subjected to the network. Interference subjected to the network was done in varying degrees; the combination of all devices running simultaneously was considered heavy, while one device operating was considered light. Our paper encompasses the testing of these interference combinations. Only under heavy interference, did the network show a 35% drop in throughput performance. Being the highest margin of performance loss, the WI-FI network still maintained an average throughput of 1.63 MB/s. The combination of two devices operating proved a 17% to 30% performance loss, while one device concluded loss ranging from 2% to 24%.

In the following section of the paper we will touch on some previous research relating to our work followed by background knowledge pertaining to wireless communication, FCC standards, and interference devices. Section 4 will cover a detailed description of our methodologies and steps taken in the experiment phase. Section 5 we display our results and analyze them. Finally, in section 6, some conclusions are presented, tying in paths towards possible future work.

2. PREVIOUS RESEARCH
Testing performance on wireless networks is a relatively new field of study. Research has been done on different aspects of WI-FI communication when dealing with interference issues. For example, interference caused by domestic microwave ovens operating in the proximity of wireless networks has been evaluated and presented by Kamerman and Erkocevic [2]. Because we used a microwave oven in our study, the results of this work are notable, but only relate to one part of our tests. In addition, studies by Jain and Padhye test the throughput performance of WLANs and the affects of interference while operating another wireless technology, Bluetooth [3]. This research relates to our work as it also tests the performance of nodes operating at close distances; however, it only focuses on

Proceedings of the 4th Winona Computer Science Undergraduate Research Seminar, April 20–21, 2004, Winona, MN, US, Copyright 2004.

Bluetooth. We are trying to show the performance based on not only throughput, but also transaction rate and response time. Additionally, we administer interference on the WI-FI network from not only single devices, but multiple devices with varying setups as well.

3. BACKGROUND

Although home wireless networks are increasingly becoming more popular, WI-FI standards maybe unknown or misunderstood. Wireless fidelity (WI-FI) is the generic term used to refer to any type of the IEEE (Institute of Electrical and Electronics Engineers) 802.11 network (including 802.11a and 802.11b) [4]. Originally, WI-FI merely referred to 802.11b, as it was the first standard to be successfully deployed on the market, but later appended its definition by adopting all 802.11 standards.

The IEEE 802.11 standard defines physical layer protocols for WLANs. The three different physical specifications are frequency hopping (FH) spread spectrum, direct sequence (DS) spread spectrum, and infrared (IR) [5]. For this study, the 802.11b standard defining the DS spread spectrum is used. The 802.11b broadcasts in an unlicensed radio band at 2.4 GHz and specification provides 11 Mbps of bandwidth. In practice, wireless networks never reach this level of performance. According to WI-FIPlanet.com, the "actual throughput you can expect to obtain from an 802.11b network will typically be between 4 and 5 Mbps" [4]. Our average throughput tests found in section 4 are slightly less than these claims due to the environment of the setup.

Knowing that the 802.11b broadcast frequency is 2.4 GHz, it is necessary to understand the band of frequencies in this range. The Federal Communications Committee (FCC) allotted the 2.4-2.5 GHz bands for industrial, scientific, and medical devices known as the ISM band. Make note that the FCC is a United States regulations committee and that frequency ranges for foreign countries may vary. ISM devices are non-radio communications devices that use radiofrequency energy for anything from heating and drying to welding [6]. Due to the nature of these devices, energy escapes, creating an unintended dissipation of radiofrequencies. In effect, the energy leaks could cause interference, especially for ISM devices generating high levels of radiofrequency energy. One of the most common examples of an ISM device is a domestic microwave oven [6], which typically operates between 2,336 MHz and 2,500 MHz. A microwave is one of the devices used in our experiments.

4. METHODOLOGY

4.1 Setup

To test the hypothesis that home WI-FI 802.11b networks falter only under heavy interference conditions, we collected a list of wireless devices to administer interference including:

- a domestic microwave oven
- a 2.4 GHz cordless telephone
- two *Wavebird*™ wireless controllers for the *Nintendo GameCube*®

The number of devices chosen is the product of available resources and amount of time given for the project. In an ideal environment, a few more devices would have been evaluated, like a Personal Data Assistant (PDA) and a wireless keyboard and mouse. To make up for a short device list, more experiments were setup and extended.

The home wireless network setup for the experiment achieved connectivity via a local (Winona, MN) broadband cable internet provider. A *Netgear*® MR814 802.11b Cable/DSL wireless router negotiated addresses for the two nodes used as well as performed the translating service to 802.11b standards. Two *Gateway*® laptops running Microsoft Windows XP and generic wireless LAN cards took the node positions and were placed in separate rooms, distancing themselves by about 15 meters. Both nodes were approximately 25 meters from the *Netgear* wireless hub and remained static throughout the experiments. A third laptop (*Gateway*®) acted as the control center for the tests, connected directly through a wall outlet wired to the cable modem. For the control center (also Windows XP) administering our performance testing, a networking performance program, IxChariot by Ixia, was chosen. IxChariot allowed for a number of tests to be run on the local network, including our three tests: throughput, transaction rate, and response time. Performance endpoints, lightweight software agents used by IxChariot, were installed on the two wireless node clients as well as the server node. Endpoints collect information about network transactions and send this information back to the console for analysis and reporting [7]. Figure 1 below depicts our setup:

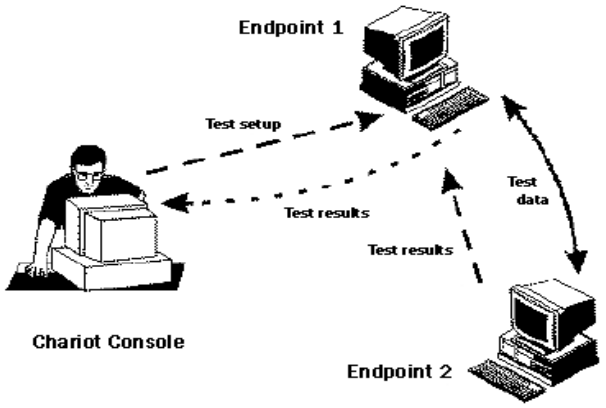


Figure 1. Endpoint and Client Setup [8]

4.2 Testing Specifics

After a moderately quick setup and a few successful performance tests with IxChariot, testing could begin. The wireless nodes' IP addresses filled Enpoint1 and Endpoint2 in IxChariot and the High Performance Throughput application script was selected. Figure 2 displays the datagram run options for the script. Window size in bytes (1500 for this test) is most significant here.

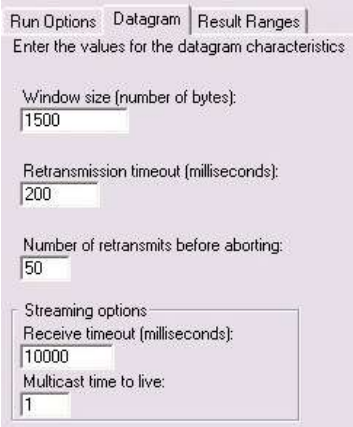


Figure 2. IxChariot Datagram run values

The throughput script allowed a few different methods of running tests, like packet count or time clock. For all the tests administered on the network, three minutes was chosen as the runtime of the script. One complete test run collected data on throughput, transaction rates measured in transactions per second, and response time in seconds. Throughput measured the relative amount of maximum bandwidth available. The transaction rate test measured the amount of time it took endpoint1 to make a single transaction with endpoint2. Response time measured the time it took the first node to complete a 10,000,000 byte transaction over five transaction periods.

The first tests were administered without interference for average throughput levels of the network. Five average tests provided a basis for comparison. Testing continued with this pattern of five tests per setup (i.e., "2.4 GHz Cordless" is one setup). A total of 40, three-minute tests were conducted in combinations of the following interference layout:

	0	**1**	**2**	**3**
None	**X**			
Microwave Oven		**X**		
2.4 GHz Cordless		**X**		
***Wavebird* Controllers (2*)**		**X**		
Microwave Oven + 2.4 GHz Cordless			**X**	
2.4 GHz Cordless + *Wavebird* Controllers			**X**	
***Wavebird* Controllers + Microwave Oven**			**X**	
Microwave Oven + 2.4 GHz Cordless + *Wavebird* Controllers				**X**

*Anytime the *Wavebird* controllers were tested, two were in use

Figure 3. Test Setup/Degrees of Interference

The microwave was in between the node positions equally at about seven and a half meters respectively. While tests were running with the microwave as an interference device, the oven heated a bowl of water for the three minute duration. The *Wavebird* Controllers were also positioned between the wireless endpoints. Controller specifications allow up to 16 channels of frequencies; however, the two controllers used were set at channels 1 and 2. The 2.4 GHz cordless had a dynamic position scheme. During tests, a local cell phone was dialed and connected while the cordless was walked around the first floor of the house. Distances ranged anywhere from 1 meter from both endpoints up to 15 meters from the nearest node. During the combination setups, the devices were administered the same as described, but in multiple forms.

5. RESULTS AND ANALYSIS

The testing of our home wireless network setup was quite successful. IxChariot produced graphs for the throughput, response time, and transaction rate tests. Examples of these graphs are located in the Appendix. Included are total of four graphs; one throughput performance test in every degree of interference (defined in Figure 3).

The results were analyzed and average values for each of the tests were calculated. Figure 5 illustrates the trend for the average throughput tests taken by IxChariot. Initial throughput was in the proximity of 2.45 MB/s with a gradual increase to about 2.6 MB/s for this scenario. The calculated average throughput, response time, and transaction rate of the wireless home network was 2.525 MB/s, 32.25 s, and .03125 transactions/s respectively. The graph in Figure 4 illustrates the percentage drop in performance of each testing scenario compared to the tests with a degree of interference of zero. The analysis of this data is broken into subsections based on degree of interference.

5.1 Degree of Interference Equaling 1

Each device producing interference was run by itself during these tests. The microwave oven dropped the performance of the network by approximately 25%; throughput, response time, and transaction rate concluded a 24%, 28%, and 20% drop respectively. The throughput rate averaged 1.94 MB/s, while the response time and transaction rate averaged 44.35 seconds and .02525 transactions/s. The network seemed to "recover" with a 50% increase in throughput after one minute of time had elapsed in every case, concluding a performance drop in the network of 15%. Figure 6 located in the Appendix shows the trend of the microwave throughput tests. In this graph, the throughput starts at 2.02 MB/s and by one minute and fifteen seconds the throughput recovered to its capped rate of 2.19 MB/s.

The *Wavebird* Controllers dropped the performance of the network by 5% in throughput, 4% in response time, and 4% in transaction rate, averaging 2.4 MB/s, 33.6 seconds, and .03 transactions/s respectively.

The 2.4 GHz cordless telephone tests affected the performance of the home wireless network the most significant in our experiments. In all cases the cordless was used, including the other degrees of interference, IxChariot timed out 2 of the 5 tests. When this occurred the wireless network completely lost signal for a 15 to 20 second time period, disabling the endpoints on the wireless nodes from communicating. Figure 7 is a screenshot of the message IxChariot produced when timing out. The time-out cases draw some cause for concern. If the importance of continual connectivity is high, a 2.4 GHz cordless phone should not be used in the home wireless network setup because of it's inconsistency in completely disrupting the signal; however, if the

importance of continual connectivity is average or low, and the 2.4 GHz cordless is not in constant use, leaving it in the setup should be fine. When just the 2.4 GHz cordless was running and it didn't time out IxChariot's testing system, throughput, response time, and transaction rate dropped 2%, 1%, and 2% respectively. The network throughput averaged 2.475 MB/s during these tests. The response time and transaction rate averaged 32.3 seconds and .03075 transactions/s.

5.2 Degree of Interference Equaling 2

During these tests, two of the devices were operating simultaneously. The 2.4 GHz cordless and microwave oven tests concluded the network performing with an average throughput of 1.7 MB/s, an average response time of 41.75 seconds, and an average transaction rate of .0205 transactions/s. The drop in performance of each of these measurements is 33% in throughput, 23% in response time, and 34% in transaction rate.

When the microwave oven and *Wavebird* Controllers were tested together, the performance of the network dropped 17% in throughput, response time, and transaction rate. Network throughput averaged 2.085 MB/s, response time averaged 38.85 seconds, and transaction rate averaged .02605 transactions/s.

The final pair of test setup—*Wavebird* Controllers and the 2.4 GHz cordless phone—illustrated a 4% drop in throughput, 2% drop in response time, and a 1% drop in transaction rate. The network performed with and average of 2.46 MB/s, 32.75 seconds, and .031 transactions/s respectively. Figure 8 illustrates the trend for the tests in this scenario. Around the one minute mark of the test, throughput performance dropped to 2.43 MB/s. By the two minute mark, the throughput had risen back to 2.49 MB/s. We feel this occurred due to the timing of our setup. The controllers were in full operation starting at about 30 seconds into each test; a 30 second delay was the result of the time taken to start IxChariot, relocate to the gaming system, and navigate to the appropriate screen where controller activity was constant.

5.3 Degree of Interference Equaling 3

The largest degree of interference included the operation of all three devices simultaneously. While the microwave oven, 2.4 GHz cordless, and *Wavebird* Controllers were operating, the network performed with an average of 1.63 MB/s in throughput, 54.65 seconds in response time, and .01975 transactions/s for transaction rate. Figure 9 illustrates the trend of throughput performance for these tests. Throughput rates started at about 1.8 MB/s and fell to a capped rate of around 1.25 MB/s. Unlike the performance "recovery" in the previous scenarios, "degree of interference 3" did not regain performance, we believe, due to the amount of activity all three devices were transmitting. Compared to the interference-free tests, a drop of 35% in throughput performance, 41% in response time, and 37% in transaction rate was noted.

6. CONCLUSIONS

Home WLANs provide flexibility while maintaining service in an environment subjected to common interference. This paper introduces some important wireless and broadcasting concepts, and then describes the setup, testing, and data analysis of our experiment on a home wireless environment. Test results, although a relatively small number (due to time and availability restrictions), shows that the network is affected only mildly by the different interference levels. Taking into consideration that a microwave and/or cordless telephone are not in use continuously, minor drops in performance with throughput and response time are not crucial. We recommend that if your wireless network requires 95% or more connectivity at all times, that your implementation does not include 2.4 GHz cordless telephones. Due to their unpredictability, causing the WLAN connection to be lost, they are a possible hindrance.

It is plausible that other wireless ISM band devices may cause interference. Each device must be tested independently to see its true effect, opening paths to future research. More devices could be collected and administered including but not limited to: keyboard, mouse, web cam, medical heart monitor, garage door opener, and PDAs. Instead of simply monitoring throughput, transaction rate, and response time of endpoint to endpoint communication, we could extend our tests to include the degree of power of each device's broadcast. Resulting trends of our results affirm that performance is affected by the power level a broadcasting device omits.

Future research may lead to more conclusive evidence about the interference impact of wireless devices, but initial studies outlined in this paper support the use of home wireless networks. Interference is minimal and quality performance (throughput, transaction rate, and response time) of the WLAN is maintained. It may be safe to say, wireless is here to stay.

7. ACKNOWLEDGMENTS

Our thanks to Jeff Maurer of *Ixia* for allowing us to utilize IxChariot on a temporary basis. For more information on IxChariot or *Ixia*, visit www.ixiacom.com.

8. REFERENCES

[1] Pastore, Michael. *Networks Going Wireless at Home and Work*. The Clickz Network, Septemeber 30, 2001. Available from: http://www.clickz.com/stats/markets/wireless/article.php/10094_888881. Access on 2004 February 5.

[2] A. Kamerman and N. Erkocevic, Microwave oven interference on wireless LANs operating in the 2.4 GHZ ISM band. In *Proceedings of the 8th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Vol. 3 (1997), 1221-1227.

[3] Jain, K. and Padhye, J. Wireless network performance: Impact of interference on multi-hop wireless network performance. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, September 2003, ACM Press, New York, NY, 66-80.

[4] Moran, Joseph. *Wireless Home Networking, Part II-V*. Nov. 2002. Available from: http://www.wi-fiplanet.com/tutorials/article.php/1497111. Access on 2004 January 24.

[5] Golmie, N. and Van Dyck, R.E. Interference evaluation of Bluetooth and 802.11b Systems. In *Wireless*

Networks, May 2003, ACM Press, New York, NY, 201-211.

[6] Australian Communications Authority. *WLANS Interference Management*. ACA Publication, July 2002. Available from: http://www.aca.gov.au/radcomm/frequency_planning/radiofrequency_planning_topics/docs/rlan-im.pdf. Access on 2004 February 5.

[7] Ixia. Endpoint Library. Available from: http://www.ixiacom.com/support/endpoint_library/. Accessed on 2004 February 20.

[8] Ixia. How IxChariot Works. Available from: http://www.ixiacom.com/support/endpoint_library/howchariotworks.php. Accessed on 2004 March 3.

9. APPENDIX

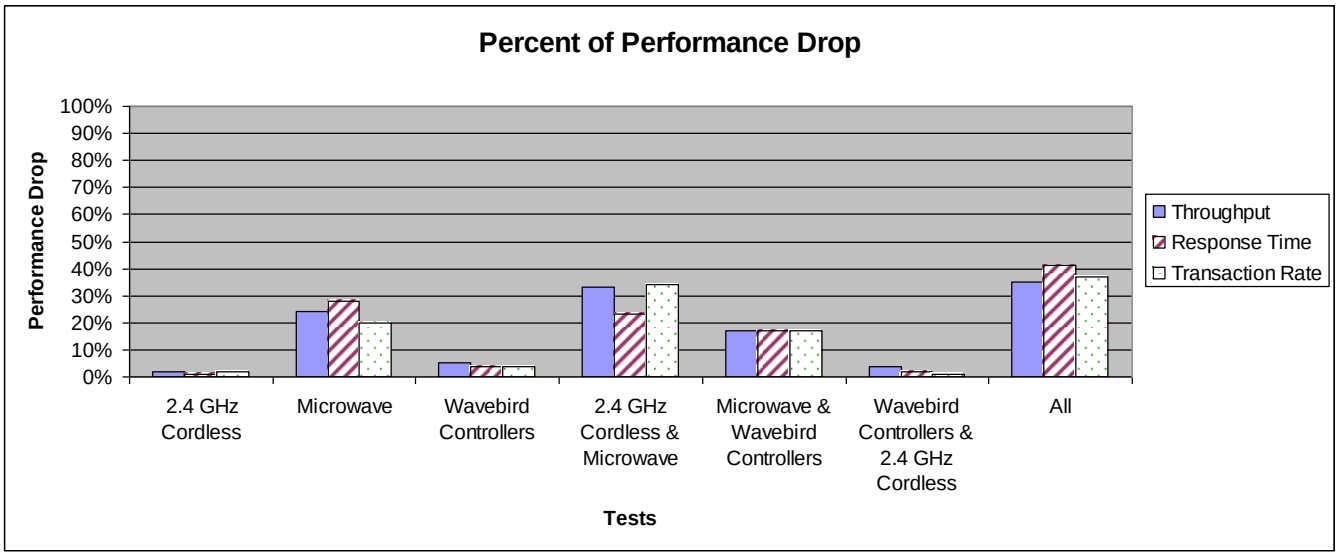


Figure 4. Percent of Performance Drop

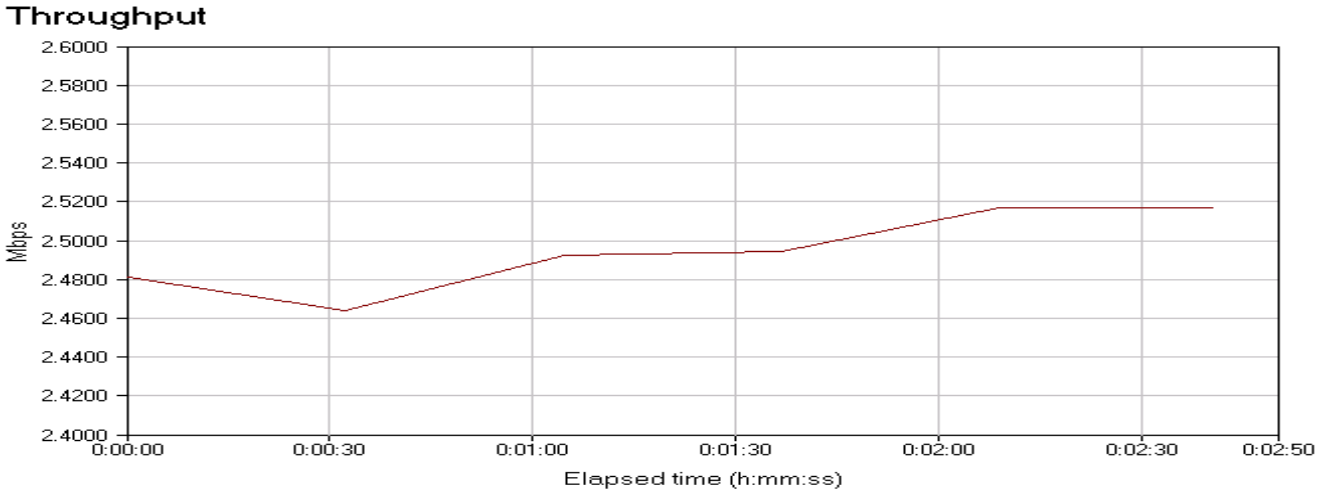


Figure 5. "Degree of Interference 0" Throughput

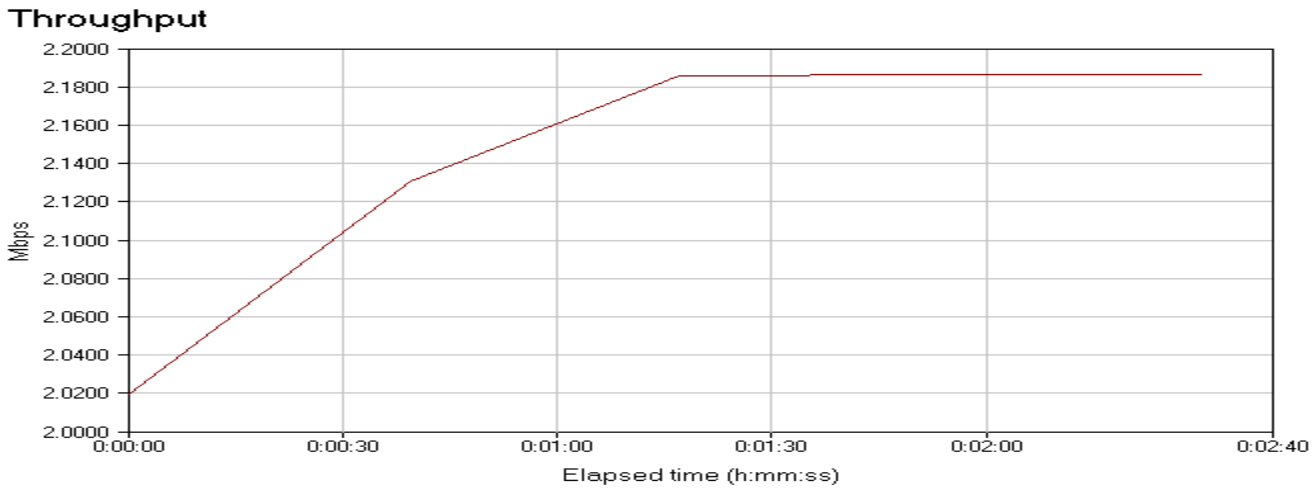


Figure 6. "Degree of Interference 1" (Microwave) Throughput

Figure 7. Timeout of 2.4 GHz Cordless

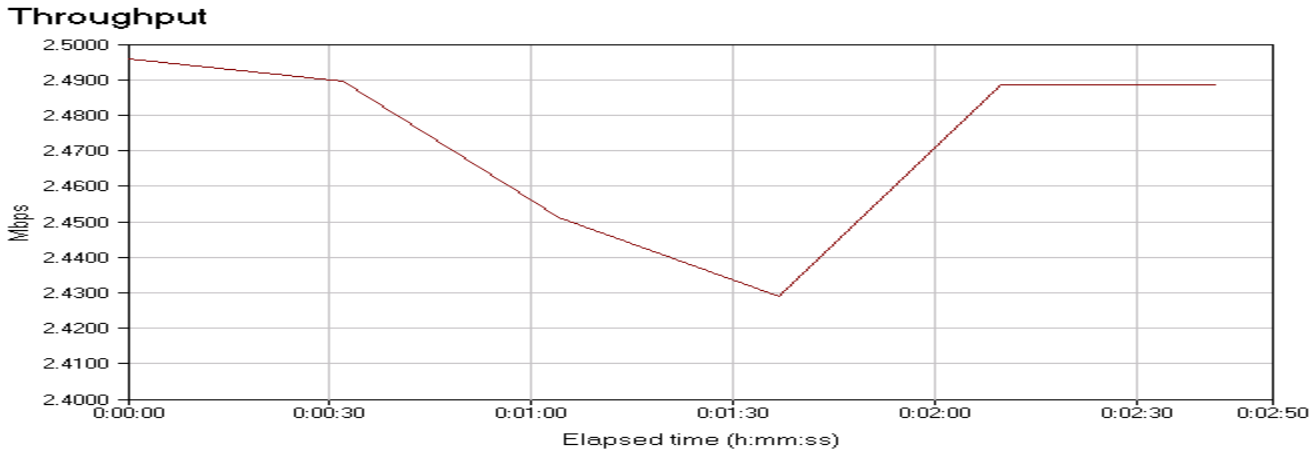


Figure 8. "Degree of Interference 2" (*Wavebird* Controllers + 2.4 GHz Cordless) Throughput

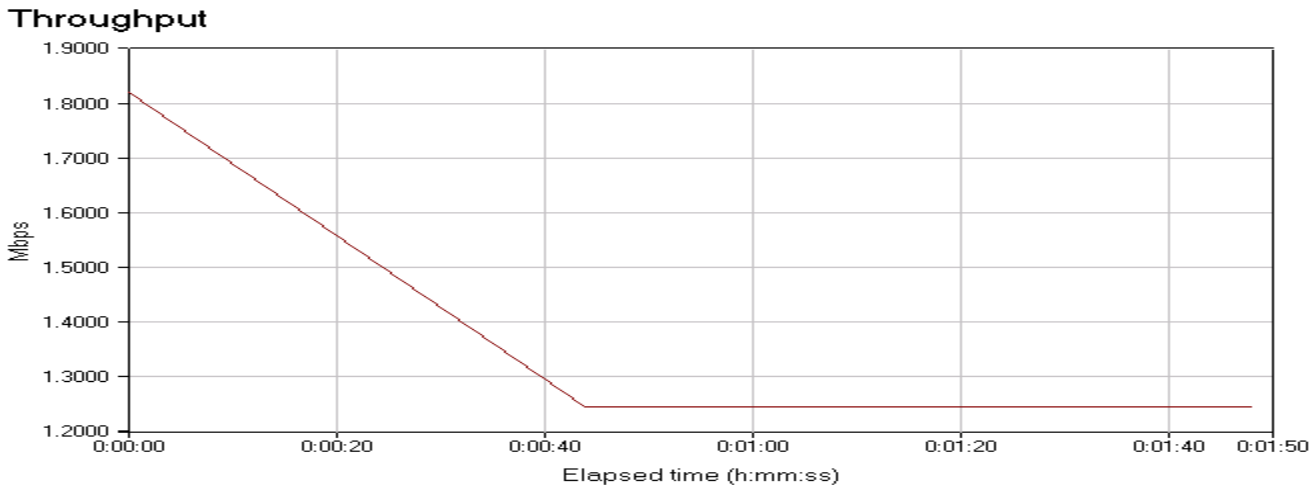


Figure 9. “Degree of Interference 3” (All 3 Devices) Throughput

Enhancing Security for Interleaving Block Cipher Modes

Charles M. Weatherhead
Saint Mary's University
700 Terrace Heights
Winona, MN 55987
cmweat01@smumn.edu

ABSTRACT

The cryptographic community is presented with numerous algorithms to encrypt data. Performance is one of the widely used concepts used to evaluate the efficiency of these algorithms. In cryptography, the speed at which a cryptographic algorithm performs is often an indicator of its usefulness. Performance, however, cannot provide the sole means of evaluation. Security is pivotal to cryptography and must be considered in turn with performance. Interleaving is one method that speeds up encryption for many algorithms. However, it could potentially hinder the security of the underlying algorithm if its restrictions are not adhered to. A new method of interleaving is proposed which aims to limit the weaknesses of typical interleaving: Single Initialization Vector Interleaving. A comparison based on encryption speeds and security analysis suggests this form of interleaving as an acceptable alternative to typical interleaving.

Keywords

Interleaving, Block Cipher Modes, Block Ciphers, Compression, Redundancy.

1. INTRODUCTION

The word *cryptography* means "secret writing". More specifically, it "is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication" [1]. Cryptography relies on *encryption* to convert plaintext to ciphertext, and *decryption* to convert ciphertext back to plaintext. *Plaintext* is the original form of a message, while *ciphertext* is the altered form of the original message. *Ciphers* are the mathematical algorithms that perform encryption and decryption. Of particular interest within this paper are *symmetric block ciphers*, which encrypt and decrypt blocks of data with the same private key.

Symmetric block ciphers are recognized as a fast means of encrypting data, but they are frequently slowed down due to various modes they may run in. A method called interleaving was introduced to counteract this effect. Its sole purpose is to enhance the encryption speed of a symmetric block cipher through parallel encryption, whereby portions of the plaintext are encrypted concurrently.

Typical interleaving requires only one key to be used by all the ciphers ran in parallel. However, a unique piece of data must also be generated for each cipher, called an *initialization vector* (IV). These IVs must adhere to certain restrictions to maintain the security of the underlying block cipher. A new method of interleaving is proposed called *Single Initialization Vector Interleaving* (SIVI), which provides for an increase in security compared to typical interleaving by limiting the use of these IVs. While enhancing security, SIVI also maintains the same encryption speed as typical interleaving.

The remainder of this paper is laid out as follows: the next section covers background research regarding the area of interleaving symmetric block ciphers. Section 3 describes the method of interleaving in more detail. A new proposed method to enhance the security of interleaving is proposed in section 4. Section 5 describes the methods employed to test the security of the new proposed method. Results and analysis are then covered in section 6. Concluding remarks are made in section 7, and section 8 suggests future work relating to this paper's new proposed method.

2. BACKGROUND RESEARCH

Symmetric block ciphers, although considerably faster than asymmetric ciphers [6], still have their limitations in performance. Such limitations come from the lack of parallelism that is able to take place [10]. This is due to the various modes a block cipher may function within. The NIST (National Institute of Standards and Technology) has published a recommendation for block cipher modes of operation [19]. Three of the five specified modes disallow the use of parallel encryption for the underlying symmetric block cipher.

In [3], a method called interleaving is introduced to remedy this limitation. Some interleaving modes have ultimately been recommended by the NIST for the triple-DES algorithm [18]. A more detailed look into the design of interleaving will be covered in the next section. A firm understanding of interleaving is needed in order to properly present a new method of interleaving addressed in section 4. A further look into the performance gains of interleaving is addressed within [20], but fails to address any potential security flaws resulting from the design of interleaving. It is this design we turn to now.

Proceedings of the 4th Winona Computer Science Undergraduate Research Seminar, April 20–21, 2004, Winona, MN, US, Copyright 2004.

3. INTERLEAVING

The following subsections will provide the basis for understanding symmetric block cipher interleaving.

3.1 Block Cipher Modes

Plaintext is often divided into blocks, typically 128 bits in size. *Block ciphers* are then able to operate on these blocks of input one block at a time. When a plaintext is able to be broken into more than one block, a *block cipher mode* is required. The sole purpose of a block cipher mode is to provide confidentiality for the underlying block cipher [16, 19]. The NIST (National Institute of Standards and Technology) has recommended block cipher modes for the cryptographic community [19]. When referring to block cipher modes within this paper, it is understood that the following modes are being regarded: ECB (Electronic Codebook), CBC (Cipher Block Chaining), CFB (Cipher Feedback), OFB (Output Feedback), and CTR (Counter).

Non-feedback cipher modes, such as ECB and CTR, are able to encrypt various blocks without relying on any previous input. *Feedback block cipher modes*, such as CBC, CFB, and OFB, rely on the encryption of a previous block before the next block may be encrypted [15].

Non-feedback modes allow for parallel encryption to take place, thus increasing the speed of encryption. *Parallel encryption* enables multiple ciphers to encrypt portions of the plaintext at the same time, rather than relying on only one cipher to work on all of the plaintext. However, non-feedback modes are unable to achieve a proper level of *diffusion*, allowing for the plaintext data to be widely dispersed over the ciphertext [6].

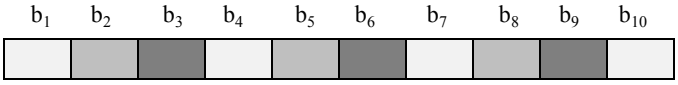
Feedback modes do not provide for parallel encryption, but they offer an increase in security due to the higher level of diffusion they supply. The following section describes a compromise that may be attained between non-feedback and feedback modes.

3.2 Interleaving Design

A method called *interleaving* combines the security of feedback modes while achieving the encryption speed of non-feedback modes [15, 20]. Rather than one chain of encryption, multiple may be created. This is achieved by chaining together various blocks of plaintext into groups. For example, the first and every third plaintext block thereafter may be grouped together as one chain. The second and every third block thereafter may be joined to form the second chain, and so on. Each chain is then able to be encrypted by its own block cipher with its corresponding block cipher mode. Figure 1 illustrates an example of how multiple encryption chains are formed when using interleaving.

When interleaving block ciphers, the same block cipher and block cipher mode is used for each encryption chain. CBC, CFB, and OFB modes are able to be used when interleaving [3]. Only one key is needed that may be used by each cipher. However, each *initialization vector* (IV) must be different for each block cipher in order to guard against the possibility of differing plaintext blocks being mapped to the same ciphertext blocks [16]. Such an occurrence would introduce patterns an attacker could exploit.

We turn now to a more thorough explanation of IVs. A further understanding of IVs is fundamental to the remainder of this paper, and is the cornerstone upon which the new proposed method of interleaving is based.



Encryption chain one: $b_1 \rightarrow b_4 \rightarrow b_7 \rightarrow b_{10}$

Encryption chain two: $b_2 \rightarrow b_5 \rightarrow b_8$

Encryption chain three: $b_3 \rightarrow b_6 \rightarrow b_9$

Figure 1. Three-way interleaving with ten blocks of plaintext.

3.3 Initialization Vectors (IVs)

3.3.1 IV Characteristics

An IV is nothing more than a "dummy" block that starts the chaining process of encryption within feedback modes. An IVs standards may vary depending upon the circumstances the IV is to operate within. Unlike the keys used in symmetric block ciphers, IVs do not need to be kept secret. There are various other characteristics that an IV may need to exhibit, depending on which block cipher mode is being used [1, 19].

An IV may need to be unpredictable, meaning the IV should not be able to be predicted before it is generated. There may also be a requirement for the IV to be unique, whereby each IV must be different from the other. The integrity of the IV may also need to be ensured, so that the IV may not be altered so as to introduce bit errors which lead to improper decryption. These characteristics must be taken into account when generating IVs.

3.3.2 IV Generation

The NIST recommends two methods for producing IVs [19]: using a nonce (a number used only once) or a PRNG (pseudorandom number generator).

A nonce requires an element of uniqueness, never allowing the same nonce to be used with the same key. There must also be enough information present within the ciphertext to allow the receiver of an encrypted message to reconstruct the nonce values used before they are able start decryption. See [16] and [19] for further details and restrictions a nonce must adhere to.

The other method allows for the IV to be generated using a PRNG. However, such generators are hard to come by [1, 16]. In fact, PRNGs are subject to attack as well [3, 11, 16]. Randomness plays a pivotal role in cryptography, and if not used properly, may weaken the cipher. When using a PRNG, it is best to not "generate too much data at one time," in order to "ensure that the data we generate will be statistically random" [16]. In fact, concerning "the majority of ciphers, it is only the existence of redundancy in the original messages that makes a solution possible" [5]. Recognizing the rigorous standards a PRNG must adhere to, the NIST has provided a test suite for PRNGs [17]. This suite employs multiple tests in order to ensure that the PRNG data is indeed cryptographically strong. The issue of redundancy and randomness and the role it plays in cryptography will be further discussed within section 5.2. Sufficient material has been covered in order to turn now to a new proposed method of interleaving.

4. SINGLE INITIALIZATION VECTOR INTERLEAVING (SIVI)

Single Initialization Vector Interleaving (SIVI) provides for many similarities between typical interleaving:

- Only operates with symmetric block ciphers
- Only one key is needed that may be used by all ciphers
- Each cipher operates on the same key size
- Each cipher operates on the same block size
- Each cipher operates in the same block cipher mode
- From the modes presented in [19], only CBC, CFB, and OFB may be used

However, there are distinct differences that are unique to SIVI:

- Only one IV is needed that may be used by all ciphers
- Each block cipher must be different from the other, therefore requiring an *n*-way SIVI to have *n* different block ciphers

Each time a typical interleaved encryption is to take place, a new set of initialization vectors needs to be produced. SIVI requires only one new IV to be produced, thereby reducing the reliance upon secure IV generation. Only one IV is needed due to the fact that each cipher employed by SIVI is distinct from the others. This removes the possibility of producing duplicate portions of ciphertext, which would happen if only one IV were used for typical interleaving. The ciphers utilized by SIVI may also remain static. Furthermore, by Kerckchoffs' principle, the ciphers chosen in SIVI may be made public [16].

For example, a three-way SIVI may specify AES, Serpent, and Twofish ciphers each in CBC mode. Each cipher has the capability of working on the same block size of 128 bits and the same key sizes of 128, 192, or 256 bits. We turn now to the proposed methods of testing an actual implementation of SIVI.

5. METHODS

5.1 Encryption Speed

SIVI claims to provide for enhanced security over typical interleaving without lessening the encryption speed. In order to properly verify that SIVI does not hinder the encryption speed compared to typical interleaving, two forms of encryption were performed using a textual plaintext value of 19.2 megabytes. Multiple multi-lingual dictionaries from [2] were combined to form this plaintext data value.

The first form utilizes three-way interleaving with its corresponding mode: AES in CBC mode. The second form executes three-way SIVI with its corresponding mode: AES, Serpent, and Twofish each in CBC mode. Each of these forms was timed in order to properly compare encryption speed. Both forms were executed five times to allow for an average encryption speed to be calculated.

The specified forms of encryption were implemented within the Java programming language, utilizing the Java Cryptography Extension (JCE) [13]. The JCE provided the underlying framework for each of the symmetric block ciphers. The AES cipher implementation is provided within the JCE. However, an additional cryptographic provider [9] was incorporated into the JCE to provide the Twofish and Serpent cipher algorithm implementations. In addition to the JCE, various other methods to perform blocking and padding were taken from [8].

In order to achieve typical interleaving and SIVI, a network of four computers was established to parallelize encryption. The client/server piece was implemented at the software level using Java. One PC acted as the server, while the remaining three PCs served as clients. Each of the clients ran only their base Windows 2000 Server operating system and version 1.4.2_03 of the Java Runtime Environment (JRE) [14] on an Intel Pentium II processor. The server PC had an Intel Pentium IV processor running the Windows XP Professional operating system and the same JRE as its clients. These PCs were all connected together through a 10 megabytes per second Netgear Ethernet Hub.

The server spawned a new thread to handle the communication between each client. The speeds recorded indicate how long it took for each thread to run, and were recorded using a Java API method call used to retrieve the system time in milliseconds from the server PC. After the execution of these encryption forms, further testing was performed.

5.2 Security

The ciphertexts resulting from each of the three encryption forms specified above was analyzed to ensure security of the encryption. This was achieved by means of compression.

Compression plays a vital role in cryptography. "The redundancy of ordinary English…is roughly 50%. This means that when we write English half of what we write is determined by the structure of the language and half is chosen freely" [4]. Compression serves to reduce any redundancies from the data being compressed [3, 7]. "If the encryption algorithm is any good, the ciphertext will not be compressible; it will look like random data [3]. Furthermore, "This makes a reasonable test of an encryption algorithm; if the ciphertext can be compressed, then the algorithm probably isn't very good" [3]. Therefore, removing any redundancy from the plaintext increases the security of the cipher [1, 3, 12]. For further detail concerning the critical role compression plays in cryptography, refer to [12].

Two data compression tools were used to compress the ciphertexts of the above forms of encryption: version 6.0 of PKZIP [21] and version 8.0 of WinZip [22]. After these tests of encryption speed and security were performed, further analysis was performed based on the results.

6. RESULTS AND ANALYSIS

6.1 Encryption Speed Analysis

Table 1 presents the average encryption speeds for each of the encryption forms specified in section 5.1, based on the same input plaintext value.

Encryption Form	Milliseconds to Encrypt
Three-way Interleaving	126,503
Three-way SIVI	126,184

Table 1. Encryption speeds for the same 19.2 megabyte plaintext.

Based on the values presented within Table 1, it is apparent that SIVI provides for essentially the same encryption speed as typical interleaving.

When evaluating the speed of SIVI, it is important to take into account the choice of ciphers. Certain ciphers are known to be computationally slower than others. For example, triple-DES is three times slower than DES [16]. Also, [20] discusses how encryption speeds may depend on how many streams of encryption exist. For example, four-way interleaving may allow for positive speedup benefits, while two-way interleaving may not.

6.2 Security Analysis

The resulting ciphertext values from each encryption form specified in section 5.1 were compressed using PKZIP and WinZip. Every file was compressed using maximum compression, but yielded no compression.

It is suggested that "the probability… [a ciphertext] file can be appreciably compressed by a general-purpose compression routine is small. (By appreciably, I mean more than 1 or 2 percent.)" [3]. Therefore, the fact that no compression could be performed on any of the encrypted files suggests that each form of encryption performed at the same security level based on compression analysis. This indicates that SIVI provides at least the same amount of security as typical interleaving, based on compression.

7. CONCLUSION

Interleaving provides the means by which symmetric block ciphers may attain encryption speeds reflective of non-feedback block cipher modes, while still providing the security inherent within feedback modes. However, with such focus centered on performance, it is believed that the reliance interleaving places on multiple Initialization Vector (IV) generation hinders the security of the underlying cryptographic cipher. Due to the inherent restrictions on IV generation, it is this paper's position that reliance on such numerous IV generations should be kept to a minimum.

Single Initialization Vector Interleaving (SIVI) provides an alternative to typical interleaving. SIVI provides an equivalent encryption speed and at least the same amount of security as typical interleaving, based on compression analysis of the resulting ciphertext values from each encryption form tested. Due to SIVIs allowance of only one IV rather than multiple, SIVI is less prone to attacks relating to IVs. It has therefore been shown that SIVI provides an acceptable substitution in place of typical interleaving.

8. FUTURE WORK

The aim of this paper was to provide the initial proposal and analysis of SIVI. Further analysis is required before SIVI may be considered trusted. There are various outlets for future work relating to SIVI.

A further investigation into the consequences of using the same key for differing ciphers should be considered. Certain ciphers may have specific key requirements that differ from other ciphers key requirements, therefore SIVI key generation should be further analyzed.

Also, a closer study could be performed on the use of multiple differing ciphers. There may be circumstances where differing ciphers may be beneficial or detrimental. A further analysis of utilizing differing ciphers needs to be undertaken to ensure that SIVI is a practical and reliable means to enhance the security of interleaving block cipher modes.

9. REFERENCES

[1] A. Menezes, P. V. Oorschot, S. Vanstone. *Handbook of Applied Cryptography*. Boca Raton: CRC Press, 1996. ISBN 0-8493-8523-7.

[2] A. Shakib-Manesh. Persian Multilanguage Dictionary v1.0 (Free). See http://www.cc.jyu.fi/~amishak/Persian_dictionary/Persian_dictionary.html.

[3] B. Schneier. *Applied Cryptography, Second Edition*: *Protocols, Algorithms, and Source Code in C*. New York: John Wiley & Sons, Inc., 1996. ISBN 0-471-11709-9.

[4] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, vol. 27, pp. 379-423 and 623-656, July and October 1948. See http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html.

[5] C. E. Shannon. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, vol. 28, pp. 656-7l5, October l949.

[6] C. P. Pfleeger, S. L. Pfleeger. *Security in Computing, Third Edition*. Upper Saddle River: Prentice Hall, 2003. ISBN 0-13-035548-8.

[7] D.A. Lelewer, D.S. Hirschberg. Data Compression. *ACM Computing Surveys*, vol.20, pp. 261-292, September 1987. See http://www.ics.uci.edu/~dan/pubs/DataCompression.html.

[8] D. Bishop. *Introduction to Cryptography with Java Applets*. Sudbury: Jones and Bartlett Publishers, 2003. ISBN 0-7637-2207-3.

[9] FlexiProvider: Harnessing the Power of the Java Cryptography Architecture ™. The FlexiProvider Group, 2002. See http://www.flexiprovider.de/.

[10] J. Burke, J. McDonald, T. Austin. Architectural Support for Fast Symmetric-Key Cryptography. In *9th International Conference on Architectural Support for Programming Languages and Operating Systems* (Asplos-IX), 2000.

[11] J. Kelsey, B. Schneier, D. Wagner, C. Hall. Cryptanalytic Attacks on Pseudorandom Number Generators. *Fast Software Encryption, Fifth International Workshop Proceedings (March 1998)*, Springer-Verlag, 1998, pp. 168-188. See http://www.schneier.com/paper-prngs.html.

[12] J. L. Massey. Some Applications of Source Coding to Cryptography. *European Trans. on Telecom*, vol. 5, pp. 421-429, July-August 1994.

[13] Java ™ Cryptography Extension (JCE) Reference Guide for the Java ™ 2 SDK, Standard Edition, v 1.4. Sun Microsystems, Inc., 2002. See

http://java.sun.com/j2se/1.4.2/docs/guide/security/jce/JCERefGuide.html.

[14] Java Technology. Sun Microsystems, Inc., 2004. See http://java.sun.com/.

[15] K. Gaj, P. Chodowiec. Hardware performance of the AES finalists - survey and analysis of results. Technical Report, George Mason University, September 2000. See http://ece.gmu.edu/crypto/AES_survey.pdf.

[16] N. Ferguson, B. Schneier. *Practical Cryptography*. Indianapolis: Wiley Publishing, Inc., 2003. ISBN 0-471-22894-X.

[17] National Institute of Standards and Technology. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. Special Publication 800-22, October 2000.

[18] National Institute of Standards and Technology. *Modes of Operation Validation System for the Triple Data Encryption Algorithm*. Special Publication 800-20, April 2000. See http://csrc.nist.gov/cryptval/des/tripledesval.html.

[19] National Institute of Standards and Technology. *Recommendation for Block Cipher Modes of Operation, Methods and Techniques*. Special Publication 800-38A, December 2001. See http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf.

[20] P. Dongara, T. N. Vijaykumar. Accelerating Private-key Cryptography via Multithreading on Symmetric Multiprocessors. *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pages 58-69, March 2003. See http://www.ece.purdue.edu/~vijay/papers/2003/icbc.pdf.

[21] Trusted ZIP Solutions for the Enterprise. PKWARE Inc., 2003. See http://www.pkzip.com.

[22] WinZip: The Zip File Utility for Windows. WinZip Computing, Inc., 2004. See http://www.winzip.com.