THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

COLLEGE OF INFORMATION SCIENCES AND TECHNOLOGY

FUTURE IMPLICATIONS OF GPU ACCELERATION ON PRESENT CRYPTOGRAPHIC STANDARDS

GARRETT MICHAEL MILLER
Spring 2011

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Information Sciences and Technology
with honors in Information Sciences and Technology

Reviewed and approved* by the following:

Stanley G. Aungst
Professor of Practice
College of Information Sciences and Technology
Thesis Supervisor

James A. Leous
Team Leader
ITS Emerging Technologies Group
Thesis Supervisor

Brian H. Cameron
Professor of Practice
College of Information Sciences and Technology
Honors Adviser

* Signatures are on file in the Schreyer Honors College.

ABSTRACT

This document surveys a number of recent developments in the information security field pertaining to parallel computing and cryptographic security, and demonstrates the performance gains made possible through the use of parallel computing in Graphics Processing Unit (GPU) utilization frameworks such as NVidia's CUDA and ATI's Stream frameworks. The NVidia CUDA framework is leveraged in a number of real world tests comparing several modern traditional central processing units (CPUs) and GPUs in the same cryptographic applications. Additional topics relevant to accelerating password cracking and cryptography are also examined, such as rainbow tables and solid state drives (SSDs), as well as cloud and distributed computing. Finally, the performance enhancements afforded by GPU parallel computing are compared against modern government and commercial cryptographic standards, and recommendations are made for retaining information security in the face of such dramatic performance increases.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

This thesis represents a culmination of my academic pursuits and interests during my time as a student in Penn State's College of Information Sciences and Technology. As such, I would like to thank the following; without them, this work could not exist:

- Dr. Stanley Aungst and Mr. Jim Leous for sharing their wisdom and volunteering their time to assist in the revision of my thesis.
- Dr. Brian Cameron for serving as my honors advisor and helping me to identify an area of research for my thesis.
- Dr. Lisa Lenze for offering her guidance and support through my time in IST.
- Dean David Hall, the College of Information Sciences and Technology, and the Schreyer Honors College for financial support of my thesis research.
- The Information Assurance Club for helping me to hone my skills and giving me the opportunity to share information security knowledge with others.
- Carl and Patricia Henninger, Peter J. Lechner, Ms. Eva Blum, and Mr. Robert Bardusch for their generous scholarship support.
- My friends and family for supporting me through this academic journey.

Introduction

In cryptography, the ability of a cryptographic function to protect the confidentiality of information has always been limited by its susceptibility to attacks against it. These attacks can vary in sophistication, but they generally require one thing in common: large amounts of computing power available to check for a large number of potential values that could exist as a "key" for a piece of encrypted information. Current standards exist in both government and industry to help ensure the confidentiality and integrity of information, but it is presently unclear if these standards have considered the dramatic performance increases made possible by leveraging the parallel computing power of Graphics Processing Units (GPUs). The designs of most cryptographic algorithms are such that they can benefit considerably from parallel computing, which consumer GPUs can provide inexpensively and economically.

NVidia CUDA, ATI Stream, and OpenCL

Overview

A recent trend in computing concerns the use of GPUs in order to perform calculations outside of the typical graphics-rendering applications for what they are intended. This technique is known as General Purpose Graphics Processing Unit (GPGPU) computing. Two of these GPU utilization frameworks include NVidia's CUDA and ATI's Stream frameworks. These frameworks are similar in that they both include development toolkits which run only on their corresponding hardware. A third open standard, known as OpenCL, provides a C-like programming environment supported by CPUs and GPUs alike, independent of manufacturer

(AMD Corporation). For the purposes of this thesis, NVidia's Compute Unified Device Architecture (CUDA) offering will be examined, as well as the potential that it has to accelerate the testing of common cryptographic functions.

According to NVidia, CUDA is "NVidia's parallel computing architecture that enables dramatic increases in computing performance by harnessing the power of the GPU" (NVidia Corporation). Whereas CPUs are very well suited to performing fewer complex serial calculations than GPUs, GPUs are designed to be well suited to computing many smaller calculations in parallel, such as rendering millions of pixels on a screen in a graphics rendering application or video game. This lends itself well to certain cryptographic applications, such as bruteforcing billions of MD5 hashes per second, or bruteforcing WPA-PSK keys. Certain prime factorization techniques used in breaking RSA moduli (*pq*) are also well-suited to GPU acceleration (Bernstein, Chen, Cheng, Lange, & Yang, 2008).

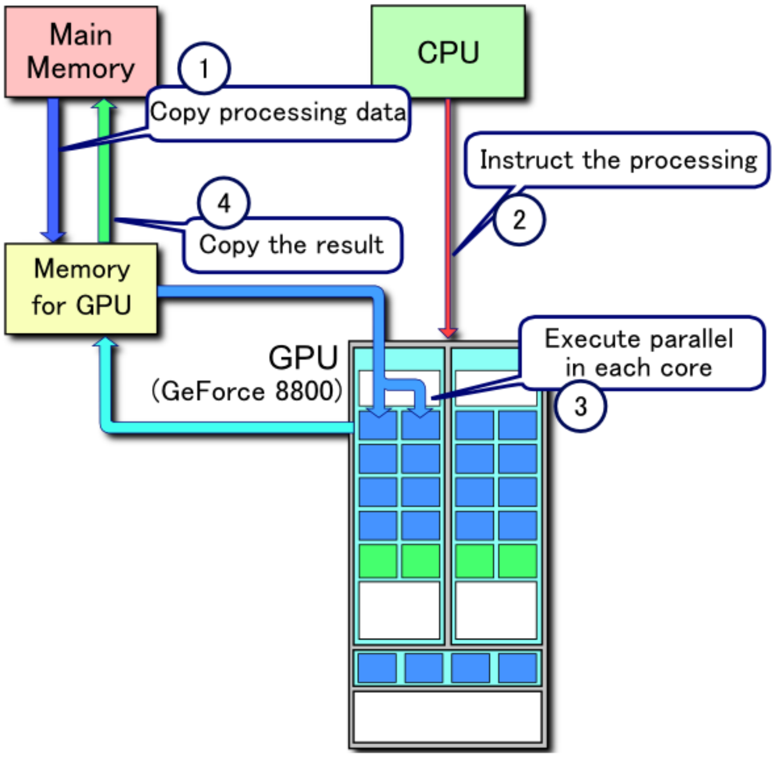

Figure 1- Processing Flow in CUDA (Tosaka, 2008).

Figure 1 details the means by which CUDA operates on the GeForce 8800 GPU. Data to be processed is copied from main memory to the GPU. The CPU then instructs the processing of the GPU, where data to be processed is loaded from GPU memory and executed in parallel on each core. Finally, the results are written back to GPU memory, where they are then copied into main system memory (Tosaka, 2008). A GTX 580 has 512 computing cores vs. an average CPU's two to four, albeit clocked lower than an average CPU. Because of the massively-parallel design of GPU hardware, applications that require many computing threads will benefit the most from GPU acceleration. Serial applications are unlikely to experience the same level of performance enhancement that parallel applications see. Another major benefit of GPU-based parallel processing is vastly increased memory bandwidth over similar CPU-based systems (Sinnott-Armstrong, Greene, Cancare, & Moore, 2009). This enables the processing cores to communicate much more rapidly with their local memory, enabling rapid calculation of large volumes of data.

History

While GPGPU computing is an area of computing that has received much attention in recent years, it is not a new idea. The use of graphics hardware for general-purpose computation dates back to the Ikonas Graphics System in 1978 (GPGPU). More recently, consumer demand for graphics hardware has fueled its development, making an extremely large amount of computing power available at a lower cost compared to that of equivalent CPU-based systems (Sinnott-Armstrong, Greene, Cancare, & Moore, 2009).

CUDA, NVidia's GPGPU solution, was introduced in 2006 with the NVidia 8000 series of GPUs. Designed to provide a C programming environment capable of running natively on graphics hardware, its purpose was to give software developers the ability to leverage parallel computing on a massive scale without the necessity of learning a new programming language (NVidia Corporation).

Cryptographic Applications of CUDA

Test Setups

For purposes of testing different cryptographic algorithms on CPU and GPU hardware, the following system setups were utilized. These system configurations are viewable in Table 1 and Table 2. Both systems ran the same version of Ubuntu Linux 10.10 64-bit, and had different, yet comparable hardware.

Table 1 – Specifications of Machines Tested

	Setup #1	**Setup #2**
Processor:	Intel Core 2 Duo E8400 @ 3.0 GHz / 3.6 GHz	Intel Pentium D E2200 @ 2.2 GHz
Motherboard:	Gigabyte EP45-UD3R	ASUS IPIBL-LB
Graphics Card:	NVidia/Zotac GTX 580	NVidia/Asus GTX 460
Memory:	8GB DDR2-800 SDRAM	3GB DDR2-800 SDRAM
Hard Drive:	G-Skill Phoenix Pro 120GB SSD	Western Digital Caviar Blue 500GB HDD
Power Supply:	XFX (Seasonic) 750W Black Edition	OCZ StealthXStream 2 600W
Operating System:	Ubuntu Linux 10.10 64-bit	Ubuntu Linux 10.10 64-bit

Table 2 - Detailed Specifications of Graphics Cards Used

	NVidia/Zotac GTX 580	**NVidia/Asus GTX 460**
Microarchitecture:	GF110 “Fermi”	GF 104 “Fermi”
CUDA Cores:	512	336
Graphics Memory:	1.5GB GDDR5	1GB GDDR5
GPU Clock Speed:	772 MHz	675 MHz
Memory Clock Speed:	1002 MHz	900 MHz
Memory Interface Width:	384-bit	256-bit
Fabrication Process	40nm	40nm

MD5/LM/NT/SHA Hashing

Background

Of all cryptographic concepts, Bruce Schneier and Niels Ferguson name hash functions as the most versatile (2003). Able to be used for encryption, authentication, or message signing, hash functions represent a vital part of many cryptographic systems. By taking an input of arbitrary length and creating a fixed-length output through a hash function, a generally-unique value is created. Because of the one-way nature of the hash function, generally, the only way to reverse the process is to calculate billions of hashes from possible guessed inputs to check for a match. For example, if a system stores user passwords as MD5 hashes, a password of “password” will hash to a value of **5f4dcc3b5aa765d61d8327deb882cf99**. Even a minor change in the input value will yield a drastically different hash – hashing “passw0rd” yields **bed128365216c019988915ed3add75fb**. This is the mark of an effective hash algorithm – the input value cannot be predicted based on a similar output. While useful for obfuscation of original text input, such as in password storage, this property of hashes is also useful for verifying integrity of data; even a single bit change or corruption in a datagram will yield a drastically different hash.

Another way to accelerate the process is through the use of rainbow tables. Rainbow tables are massive collections of pre-computed hash values that can be many gigabytes or terabytes in size. By performing this sort of time/space tradeoff, particularly when combined with the fast read speeds of a solid-state drive (SSD), Windows XP passwords of up to 14 characters, including upper/lowercase letters, numbers, and special characters could be cracked in an average of five seconds (Objectif Sécurité, 2010).

Available Tools

Because of their ubiquitous nature in cryptography and computing in general, there is a wealth of utilities available to attack MD5 and other types of hashes. Some of these utilities are enumerated below:

- CUDA Multiforcer (GPU)
- GPU MD5 Crack (GPU)
- Hashcat (CPU)
- oclHashcat (GPU)
- John the Ripper (CPU)
- Rainbowcrack (CPU/GPU)

Testing – CPU/GPU Comparison

To better compare the performance advantages afforded by the GPU architecture vs. CPU architecture, a number of tools were tested on both CPU and GPU platforms. Table 3 reveals a number of interesting facts – that advancement of the CUDA Multiforcer from version 0.72 to version 0.80a brings demonstrable performance enhancements, reflecting the rapidly-changing nature of the industry, but the oclHashcat and hashcat tools remain the fastest of the set. Between the GTX 580 and stock-clocked E8400 at 3.0 GHz, a nearly 97x performance increase is observed.

For a random 8-character password of upper and lower-case alphabetic and numerical composition (62 possible characters), there exist 62^8 = 218,340,105,584,896 possible password combinations. Time to exhaust this keyspace on a single GTX 580 could be found by:

218,340,105,584,896 combinations / 2,196,400,000 hashes per second =

99408.17 seconds / 60 = 1656 minutes / 60 = 27.6 hours /24 = 1.15 days.

This is easily attainable. Clustering multiple GTX 580s or similar graphics cards together would make this task even more trivial, and would enable the attacking of even larger keyspaces. To complete this same task on the stock-clocked E8400 would take 111.37 days. Overclocking the processor to 3.6 GHz would only accelerate this to 90.28 days.

Table 3 - MD5 Search Speed by Device in Millions/Second

Test	**GTX 580**	**GTX 460**	**E8400@3.6**	**E8400@3.0**	**E2200@2.2**
CUDA Multiforcer 0.72	466.5	179.8			
CUDA Multiforcer 0.80a	1167.75	464			
oclHashCat .26b	2196.4	884.5			
Rainbowcrack rtgen 1.5			9.34	7.83	4.53
hashcat 0.36			27.99	22.69	16.74

RSA (Rivest, Shamir, Adleman)

Background

In 1977, Ron Rivest, Adi Shamir, and Leonard Adleman of the Massachusetts Institute of Technology (MIT) developed the RSA public-key cryptosystem. The system was published in Communications of the ACM (Rivest, Shamir, & Adleman, 1978) in February 1978, and was the first known encryption standard to introduce the concept of digital signatures (Vacca, 2009). It later became known through declassification of documents that cryptographer Clifford Cocks of the U.K. Government Communications Headquarters (GCHQ) developed a cryptosystem utilizing the same process, but due to its classified nature and computational infeasibility of the day, it was never adopted (Vacca, 2009).

Today, when securing information for transmission over a network or other insecure communications channel, RSA is the most widely used standard for public key cryptography (Schneier & Ferguson, Practical Cryptography, 2003). RSA is a public key, asymmetric cipher, which uses separate keys for encryption and decryption of messages – each user having a pair of keys, one public, and one private. Able to be used for both signing and encryption of messages, RSA ensures that nobody but the intended recipient will be able to read a message, and that the reader will be able to verify the original sender of the message. Of the "triangle" of information security - Confidentiality, Integrity, and Availability, RSA is able to offer both data confidentiality and integrity, and a certain degree of nonrepudiation, as well. RSA, however, like all other computer security concepts, is not impervious to attack.

The RSA algorithm works in the following manner:

Key Generation:

1. Select two extremely large prime numbers, p and q.
2. Calculate public key pq.
3. Select encryption exponent, in most cases, e=3 is sufficient.
4. Calculate private key $d = e^{-1} \bmod (p - 1) * (q - 1)$

This key generation gives the keys necessary to begin the RSA encryption process. A user's public key is represented by $\underline{pq}$ and e, and the private key is d. It is imperative to keep d secret, as the security of RSA is entirely dependent on keeping this number secret. If d is known, the message can be decrypted. The other processes used in RSA are detailed as follows:

Encryption:

1. Take message m
2. Calculate $c = m^{\mathrm{e}} \bmod pq$
3. Ciphertext = c

Decryption:

1. Take ciphertext c
2. Calculate $m = c^{\mathrm{d}} \bmod pq$
3. Message = m

Signing:

1. Sender hashes message to h
2. Sender calculates signature $s = h^{\mathrm{d}} \bmod pq$

3. Sender appends signature to message.
4. Receiver takes signature, calculates $h = s^e \bmod pq$
5. If hashes match, message is authentic.

The security of RSA encryption lies in the inherent difficulty of factoring extremely large numbers. If one were able to find the prime factors of *pq*, one would be able to compute or find the encryption key *e*, and its modular inverse, *d*. With the recent factorization of an RSA-768 modulus (Kleinjung, et al., 2010), one must consider implementations of RSA-1024 and their security relative to what may be computationally possible in the near future.

The best-performing factorization techniques are, in order of fastest to slowest, the General Number Field Sieve, the Multiple Polynomial Quadratic Sieve, and finally, the Lenstra Elliptic Curve Factorization Method (ECM) algorithm. In 2008, Bernstein et al. developed a GPU-based implementation of prime factorization using the elliptic curve method of integer factorization.

Bernstein et al. found that two modern NVidia GTX 295 graphics cards using their ECM factoring implementation were able to calculate 801.4 curves/second vs. the 124.71 curves/second made possible by using all four cores of an Intel Core 2 Quad Q6600 CPU (2008). They go on to state that on a single GTX 295, the implementation performs 41.88 million modular multiplications/second vs. 13.03 on a Q6600 (2008). Kleinjung et al.'s factorization of the RSA-768 modulus was completed on conventional CPU hardware. By leveraging the computational power of GPUs, it is possible that RSA-1024 could be threatened much sooner than initially anticipated. Many of the same people involved in the RSA-768 factorization assessed risk to RSA-1024 to be small until 2014 (Bos, Kaihara, Kleinjung, Lenstra, & Montgomery, 2009), though this recommendation was made prior to the RSA-768 factorization. Bernstein has also encouraged the idea of leveraging machines utilizing both CPU and GPU resources to break RSA-1024 in a talk given at the University of Illinois at Chicago (n. d.).

Available Tools

Bernstein, Cheng, et al. have developed a number of prime factorization implementations for 64-bit x86 systems. At the time of writing, however, these implementations were not able to be successfully compiled and run.

- EECM (CPU)
- GMP-ECM (CPU)
- CUDA-EECM (GPU)
- GPU ECM (GPU)

Testing – CPU/GPU Comparison

At time of writing, the CUDA-EECM software was unable to be compiled on the author's computer. This will become a point of future work. However, Bernstein, et al.'s results are displayed in Table 4.

Table 4 - Comparison of ECM Implementations

Test	GTX 295	Q6600 @ 2.4 GHz
GMP-ECM		124.71 curves/sec, 280-bit integers
GPU ECM	801.4 curves/sec, 280-bit integers	

As found by Bernstein, et al. (2008)

Elliptic Curve Cryptography

Background

As available computing power increases, the security of public key cryptography as it stands is being threatened with the recent factorization of a 768-bit RSA modulus. Key lengths can continue to grow indefinitely to maintain a semblance of security, or more efficient, more effective algorithms can be considered. The Canadian company Certicom, Inc. holds more than 130 patents related to the fields of Elliptic Curve Cryptography (ECC) and public-key cryptography, which is viewed by many in academia and industry as a roadblock to widespread adoption and implementation (National Security Agency, 2009). In an attempt to make ECC available for widespread implementation, the National Security Agency (NSA) licensed the entirety of Certicom's intellectual property under restricted terms of use. The license includes restrictions relating to suitable applications (namely, national security), as well as specific cryptographic parameters of the algorithm. Additionally, the NSA also licensed the right to sublicense this intellectual property to vendors supplying equipment within the purview of this field of use (National Security Agency, 2009).

Consequently, the NSA has endorsed ECC as an integral part of its Suite B set of cryptographic standards (NSA Suite B Security, 2009). NSA has also published a set of National Institute of Standards and Technology (NIST)-recommended key sizes detailing equivalent security levels of symmetric (AES) encryption, as well as public-key encryption of both RSA-Diffie Hellman and Elliptic Curve varieties, viewable in Table 5.

Table 5 - NIST Recommended Key Sizes

Symmetric Key Size (bits)	**RSA and Diffie-Hellman Key Size (bits)**	**Elliptic Curve Key Size (bits)**
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

(National Security Agency, 2009)

As Table 5 illustrates, despite RSA's ubiquity, elliptic curve cryptosystems offer equivalent security at much smaller key sizes compared to their older counterparts. Combined with this greater efficiency and NIST/NSA Suite B endorsement, ECC appears to be an indispensible part of the immediate cryptographic future.

Available Tools

Presently, no known utilities can be publicly found to crack ECC cryptosystems. Nevertheless, a group of cryptographic researchers have evaluated a number of different platforms for breaking ECC in response to Certicom's published ECC challenges. Initially, an NVidia GTX 295 GPU was shown to be only marginally faster than a Core 2 CPU (Bailey, et al., 2009), but later code optimizations behind an anonymous, public, collaborative effort to break ECC2K-130 have pushed the GTX 295 GPU to 54.03 million iterations of the ECC implementation per second, surpassing field-programmable gate array (FPGA)-based systems, the PlayStation 3's Cell CPU, as well as general-purpose x86-based CPU systems (Breaking ECC2K-130).

Testing – CPU/GPU Comparison

Breaking ECC2K-130 takes an average of $2^{60.9}$ iterations (Bailey, et al., 2009). Based on these latest numbers, a cluster of 1,263 GTX 295 GPUs could complete this task in one year. To accomplish this same task would require 2,026 Spartan-3 FPGAs, 2,466 PlayStation 3 Cell CPUs, or 3,039 Core 2 Extreme Q6850 CPUs (Breaking ECC2K-130). Only a purpose-built application-specific integrated circuit (ASIC) unit could surpass this GPU implementation, again showing the distinct advantage that GPUs hold over conventional CPUs in many cryptographic applications.

WEP/WPA

Background

When the 802.11 standard for wireless LAN communications was first published, it included provisions for a method of protecting wireless traffic sent across the network. Dubbed "Wired Equivalent Privacy" (WEP), it used the RC4 stream cipher to encrypt communications across an 802.11 wireless network (IEEE Computer Society, 2007). The RC4 cipher itself is not broken, but the original standard specified a key length of only 40 bits (Vacca, 2009), and required the reuse of a short initialization vector (IV), resulting in many packets being encrypted using the same key stream (Schneier & Ferguson, 2003). In order to ascertain security of a stream cipher such as RC4, the use of a unique "nonce" – a number used only once, is required to prevent cryptanalysis which can reveal the original key (Schneier & Ferguson, 2003).

In response to this fatally-flawed standard, Wi-Fi Protected Access (WPA) and its successor, Wi-Fi Protected Access 2 (WPA2) were introduced as part of the Institute of Electrical and Electronics Engineers (IEEE) 802.11i process (Frankel, Eydt, Owens, & Scarfone, 2007). In an attempt to secure vulnerable networks before the final IEEE 802.11i standard was ratified, however, the Wi-Fi Alliance introduced their own version of the WPA standard based on drafts of 802.11i. The IEEE wished to include support for the Federal Information Processing Standards (FIPS)-approved Advanced Encryption Standard (AES) as part of the WPA standard, but the Alliance did not include this, citing hardware computational capability concerns (Frankel, Eydt, Owens, & Scarfone, 2007). FIPS-approved AES was later added as part of the WPA2-supporting, 802.11i final standard. In this way, WPA was like an early stopgap measure for the vulnerabilities inherent in WEP, whereas WPA2 represents the final 802.11i standard as the IEEE and industry intended. The Payment Card Industry (PCI) Security Standards Council, recognizing these weaknesses (albeit a bit late) has prohibited the use of WEP for payment processing systems as of 30 June 2010, as stated in their PCI-Data Security Standard (PCI-DSS) version 2.0 document (Payment Card Industry Data Security Standard v2.0, 2010).

WEP/RC4 cracking remains a computationally trivial task, as shown by FBI agents being able to crack 128-bit WEP in less than three minutes at an Information Systems Security Association conference in 2005 (Cheung, 2005). As such, it is unlikely to benefit from GPU acceleration. One method of attacking a WPA/WPA2 network with a pre-shared key (PSK) is to sniff an association packet, capture the handshake that occurs, and attempt to calculate the correct pairwise master key (PMK) to determine the original PSK. Once this pre-shared key is discovered, a malicious attacker can then connect to the network, where a number of other attacks such as eavesdropping, man-in-the-middle, or traffic redirection could be executed.

Available Tools

As the popularity and ubiquity of 802.11-based wireless networks grew, so did tools to attack them. These tools do not exploit any critical weakness in the WPA or WPA2 encryption ciphers, but are able to run either bruteforce or dictionary attacks on a captured WPA/WPA2 association packet. Two such tools are listed below:

- Pyrit (CPU/GPU)
- Aircrack-ng (CPU)

Testing – CPU/GPU Comparison

While WEP cracking is computationally trivial, calculating many WPA keys remains a challenge. In order to determine the potential for speedup, Pyrit v0.4 was tested on all devices available.

Table 6 - WPA PMK Search Speed per Second by Device

Test	GTX 580	GTX 460	E8400 @ 3.6 GHz	E8400 @ 3.0 GHz	E2200 @ 2.2 GHz
Pyrit 0.4	36156.5 PMK/sec	14410 PMK/sec	2015 PMK/sec	1665.7 PMK/sec	1254.72 PMK/sec

As Table 6 illustrates, the GPUs again have a distinct advantage over the CPUs in this test. The GTX 580 outperforms even the overclocked E8400 by nearly 18 times. Pyrit also has the ability to leverage the computational resources of a machine's available CPU and GPU hardware simultaneously, as well as the hardware available on any network node also running Pyrit. As such, the potential that Pyrit holds for being used in massively distributed implementations is vast, with the combined resources of both CPUs and GPUs on many different machines being able to contribute to the cracking task. For purposes of isolating individual device performance, however, individual devices were selected for these Pyrit benchmarks.

AES (Rijndael)

Background

A current, officially-endorsed, (FIPS, NSA) cryptographic standard for symmetric encryption is the Advanced Encryption Standard (AES). Also known as the Rijndael cipher, it is one of the most popular current encryption standards in use today (Vacca, 2009).

The AES standard came to be through a NIST competition. Fifteen original standards were proposed, with ten being eliminated in the first round (Schneier & Ferguson, Practical Cryptography, 2003). Ultimately, the Rijndael cipher was accepted as the final standard for AES, with established key sizes of 128, 192, and 256-bits (FIPS 197 - Announcing the Advanced Encryption Standard, 2001). A part of NSA's Suite B cryptographic standards, NSA deems AES-

128 suitable for protecting information up to the SECRET level, and AES-256 suitable for TOP SECRET information (NSA Suite B Security, 2009).

The mathematical details of the cipher are beyond the scope of this document, but consist of ten to fourteen rounds of substituting bytes, shifting rows, mixing columns, and XORing bits (FIPS 197 - Announcing the Advanced Encryption Standard, 2001). The result is an incredibly strong, yet efficient symmetric-key cipher with key sizes much smaller than their public-key counterparts, such as RSA (National Security Agency, 2009). Because of the relative efficiency of symmetric-key ciphers, public-key ciphers will often be used only for a key exchange for a symmetric cipher to encrypt the session, reducing both CPU and network overhead (The Case for Elliptic Curve Cryptography, 2009).

Available Tools

A wide variety of tools implement the AES cipher, its standardization most certainly lending to its widespread adoption. Microsoft Office, 7zip, WinZip, and WinRAR all implement the AES cipher. Tools to attack AES-encrypted archives include:

- cRARk (CPU/GPU)
- RAR GPU (GPU)
- Rarcrack (CPU)

Testing – CPU/GPU Comparison

In order to examine the rates at which AES passphrases could be tested, cRARk 3.3c was used with a cRARk-supplied 1.2KB AES-encrypted file.

Table 7 – RAR-AES Passphrase Search Speed by Device

Test	GTX 580	GTX 460	E8400 @ 3.6 GHz	E8400 @ 3.0 GHz	E2200 @ 2.2 GHz
cRARk 3.3c	6289 pass/sec	3139 pass/sec	298 pass/sec	251 pass/sec	178 pass/sec

Again, the GPUs decisively outperform the computing power available in the CPUs in this scenario, as shown in Table 7. Of course, these numbers are miniscule in comparison to the *billions* of MD5s that could be computed each second, which serves as a testament to AES' complexity, lending to its overall security. While GPUs are able to accelerate the process considerably, the process is still rather slow compared to computing MD5 hashes. Time to exhaust even the six-character alphanumeric keyspace can be found by:

62^6 = 56,800,235,584 combinations / 6,289 passphrases per second =

9031680 seconds / 60 = 150528 minutes / 60 = 2508 hours /24 = 104 days

While adding cracking power through the use of additional GPUs or machines could increase speeds in a linear fashion, adding length or complexity to the passphrase (through adding special characters) could grow the complexity of this problem exponentially. As such, AES appears to be secure against GPU-based attacks, but only when a secure passphrase is used. A six-character alphanumeric password is not an example of a strong passphrase.

Leveraging Clusters and the Cloud

As mentioned, clustering multiple GPUs or even clusters of GPU-based cracking machines could increase cracking power in a linear fashion. This will be insufficient to keep up with the exponential growth of encryption complexity as passphrase and key sizes increase.

Nevertheless, a number of individuals, such as David Kennedy of SecManiac.com, have constructed their own GPU-based computing clusters (2011). Currently based on eight NVidia GeForce GTX 580 graphics cards, Kennedy's cluster is able to compute 14.2 billion MD5 hashes per second (Kennedy, 2011). Compared to the author's tested 2.19 billion MD5 hashes per second on a single GTX 580, it would appear that these cards scale with approximately 81% efficiency. Kennedy expects his cracking power to increase with further software optimizations and the addition of more hardware (2011).

Another option for password cracking on a large scale is leveraging cloud services, such as Amazon's Elastic Compute Cloud (EC2) service. Amazon EC2 allows users to dynamically launch server instances in the cloud with a number of possible software and hardware configurations available (Amazon Elastic Compute Cloud). Among these configurations are high-performance computing server instances which come with GPU hardware and the corresponding utilization frameworks. Robert Imhoff of Atheros Communications, Inc. has produced a calculator to determine the efficacy and cost-effectiveness of local GPU-based passphrase cracking vs. leveraging Amazon's EC2 system (2011), which he debuted at information security convention Shmoocon 2011. Imhoff's calculations reveal a great deal about GPU-based password cracking, such as how large of a keyspace a particular passphrase will produce, the time it will take to crack based on a variety of system configurations, and finally, whether it is more cost effective to build a GTX 570-based GPU cluster to crack it, or lease computing power from Amazon. Based on Imhoff's calculations, it is generally more cost effective to build a cluster if there are multiple passphrases to be cracked, as this represents a one-time cost, vs. recurring usage charges incurred by using Amazon EC2. Nevertheless, Amazon EC2 represents one way to gain access to an immense amount of computing power in a very short amount of time.

Future Work

While the results shown in this document demonstrate that GPUs have a distinct advantage over CPUs in many cryptographic applications, there remains work to be done. While Bernstein, et al's work in ECM prime factorization on GPUs demonstrated that the GPU (NVidia GTX 295) outperformed the CPU (Intel Core 2 Quad Q6600) used in the tests, these results are dated (2008). The author was unable to compile this software for testing on a more modern GPU at the time of writing, though work will continue on this. Additionally, as shown in Table 3, just a small increase in version number of the CUDA Multiforcer dramatically increased performance, so it is likely that code optimizations of GPU implementations will bring further performance enhancements, which should be examined further in the near future.

Implications on Present Cryptographic Standards

Current Payment Card Industry (PCI) Standards

To protect consumers and companies alike from credit card fraud, the Payment Card Industry Security Standards Council has published a series of Payment Card Industry Data Security Standards (PCI-DSS) documents. These documents outline a number of information security requirements surrounding card-based payment processing and customer information handling and storage.

The cryptographic requirements of PCI-DSS 2.0 call for "strong cryptography", which the standard defines as:

- Advanced Encryption Standard (AES) – 128 bit or greater
- Triple DES or 3DES – Double-length keys or greater
- Rivest, Shamir, Adleman (RSA) – 1024 bits or greater
- Elliptic Curve Cryptography – 160 bits or greater
- ElGamal – 1024 bits or greater

This standard may not necessarily be the ideal model of information security, however, as PCI-DSS only prohibited the use of WEP for securing wireless transmissions as of 30 June 2010 (Payment Card Industry Data Security Standard v2.0, 2010). In order to gain another perspective on present encryption standards, the federal government's cryptographic endorsements will now be examined.

Current NIST/NSA Standards

In order to protect information considered vital to national security and ensure software and hardware interoperability, the National Security Agency has established a set of cryptographic algorithms known as Suite B (NSA Suite B Security, 2009). Announced in February 2005, Suite B outlines protocols and algorithms deemed suitable for use, as well as prescribed key lengths and moduli for information of varying security levels.

Suite B specifies these algorithms for the following tasks:

- **Encryption:** Advanced Encryption Standard (AES) – 128/256 bits
- **Key Exchange:** Elliptic Curve Diffie-Hellman (ECDH) – 256/384 bit prime moduli
- **Digital Signature:** Elliptic Curve Digital Signature Algorithm (ECDSA) – 256/384 bit prime moduli
- **Hashing:** Secure Hash Algorithm (SHA) – 256/384 bit

The majority of public-key cryptosystems are currently set up with 1024-bit parameters, which U.S. NIST recommended was sufficient until 2010. Now that this endorsement of RSA-1024 has expired, NSA Suite B permits the use of RSA-2048 for protection of US Government information up to the SECRET level until full Suite B compliance can be achieved (National Security Agency, 2009).

Conclusions

Based on the findings herein, it can be reasonably concluded that the rapid ascent of low-cost computing power available in GPUs does threaten to force reconsideration of current cryptographic mandates. The recent expiration of NIST's RSA-1024 endorsement, combined with Bos, et al's assessment that risk is small "until 2014" would appear to be a final nail in the coffin for RSA-1024 (2009). It remains in widespread use, however, and is a standard currently endorsed by the Payment Card Industry Security Standards Council. The PCI should reevaluate this endorsement, and begin to phase out the use of RSA-1024 for payment processing and related tasks immediately. The computing resources necessary to crack RSA-1024, ECC2K-130, or moderately-weak AES passphrases may not be presently available or attainable by an individual. Nevertheless, governments or cooperative public cracking clusters could feasibly combine this amount of computing power in a relatively short amount of time. The only way to ensure proper security moving forward is to employ the same information security practices that have been echoed for years, but to exercise even more vigilance and to be unyielding in their implementation. MD5 can be bruteforced too quickly. RSA-1024 stands to be threatened soon. ECC2K-130 is also well within reach of any entity with sufficient resources, and 160-bit may soon follow. Finally, AES and WPA's protection can be ensured only through the use of sufficiently strong passphrases.

Based on this technology still in its nascent stages, more secure hash algorithms such as SHA-256, SHA-384, or SHA-512 with larger keyspaces should be implemented whenever possible. The U.S. Government has already adopted this standard, and industry and everyday users' implementations should also follow. With the threatened security of RSA-1024, RSA should be used with no less than 2048-bit moduli, and ECC 224 or 256-bit key sizes (as per NIST recommendations). AES and WPA remain secure, but again, only when paired with sufficiently strong passphrases. Microsoft's Jesper Johansson has recommended that for passphrases to be sufficiently strong, they should be physically written down (Kotadia, 2005). Johansson evaluates the risk of passphrase reuse or weak passphrases by users to be greater than the inherent risk of having passwords written down in a secure location. Cryptographer Bruce Schneier has agreed with Johansson's recommendations (Schneier, 2005). Finally, on password-based systems, multifactor authentication should be employed wherever possible. Historically considered expensive, Google has recently released a free, open source solution, compatible with any application which supports Pluggable Authentication Modules (PAMs) (Google Authenticator).

Even so, the bruteforcing of passwords is not necessarily the best point of entry into an information system. Breaking the aforementioned ciphers requires a great deal of time and resources, which an attacker is unlikely to invest if there exists an easier means of gaining access. Exploiting human factors, through the use of social engineering (getting a target to inadvertently reveal their system credentials) will remain the most easily exploitable component of a given information system. Beyond exploiting the user directly, hardware or software keyloggers also render even the most secure passphrases useless. Even as the equipment used in cryptographic research has changed, it appears that the same core philosophies have not. Strong passwords, strong algorithms, and user education remain the pillars of information security. One must fully consider what information needs to be protected, what stands to be lost if it is compromised, and take suitable measures to ensure that this information is appropriately protected.

Appendix A – CPU vs. GPU Test Results in Tabular Format

Table 8 - Brute Force Speeds by Test and Device

Test	Type	GTX 580	GTX 460	E8400 @ 3.6 GHz	E8400 @ 3.0 GHz	E2200 @ 2.2 GHz
CUDA Multiforcer 0.72	MD5	466.5 M/sec	179.8 M/sec			
CUDA Multiforcer 0.80a	MD5	1167.75 M/sec	464 M/sec			
oclHashcat 0.26b	MD5	2196.4 M/sec	884.5 M/sec			
Rainbowcrack rtgen 1.5	MD5			9.34 M/sec	7.83 M/sec	4.53 M/sec
hashcat 0.36	MD5			27.99 M/sec	22.69 M/sec	16.74 M/sec
cRARk 3.3c	RAR-AES	6289 pass/sec	3139 pass/sec	298 pass/sec	251 pass/sec	178 pass/sec
Pyrit 0.4	WPA	36156.5 PMK/sec	14410 PMK/sec	2015 PMK/sec	1665.7 PMK/sec	1254.72 PMK/sec

Appendix B – CPU vs. GPU Test Results in Graph Format

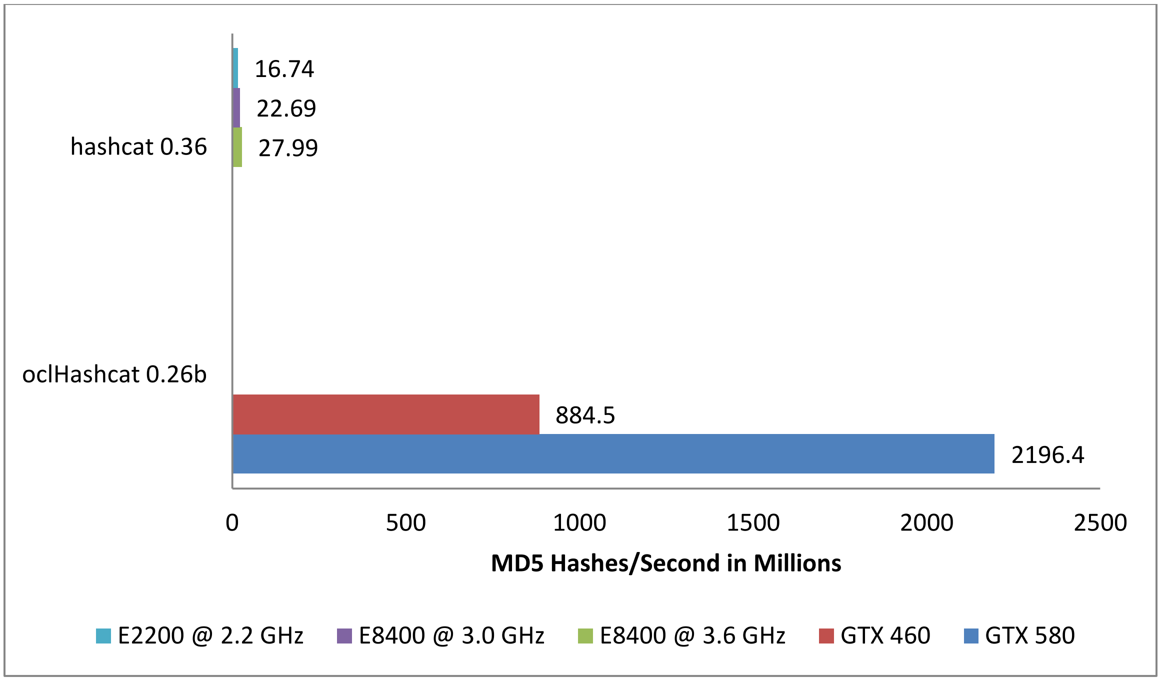

Figure 2- MD5 Hashes in Millions per Second by Device

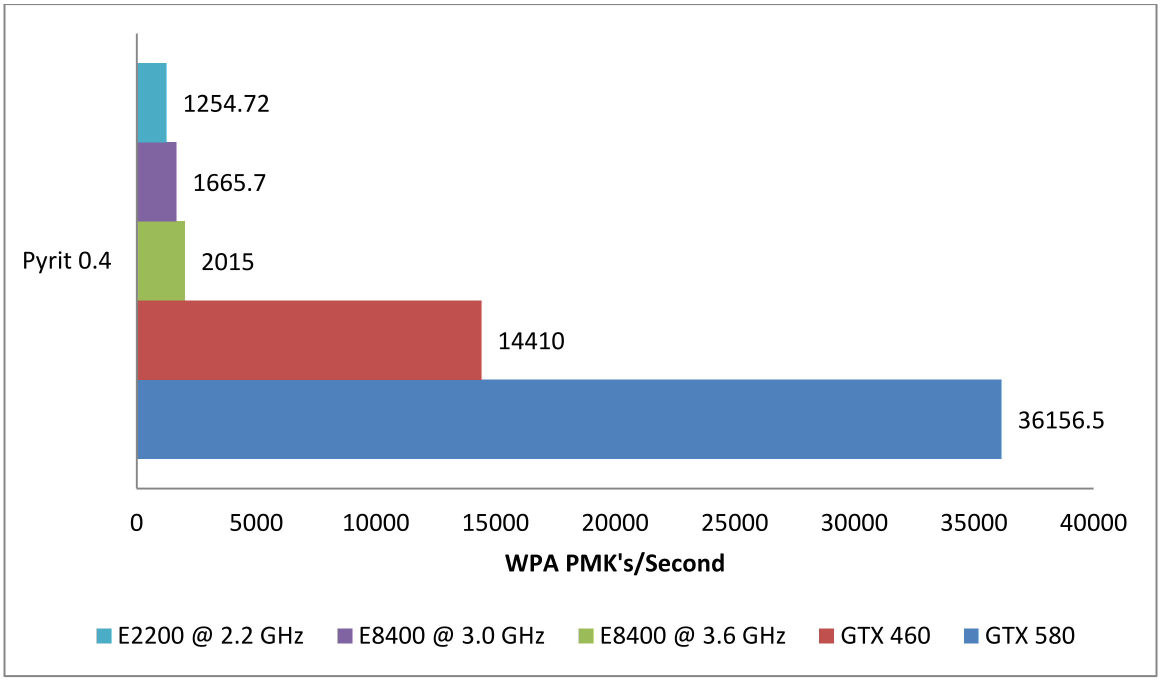

Figure 3 - WPA Pairwise Master Keys per Second by Device

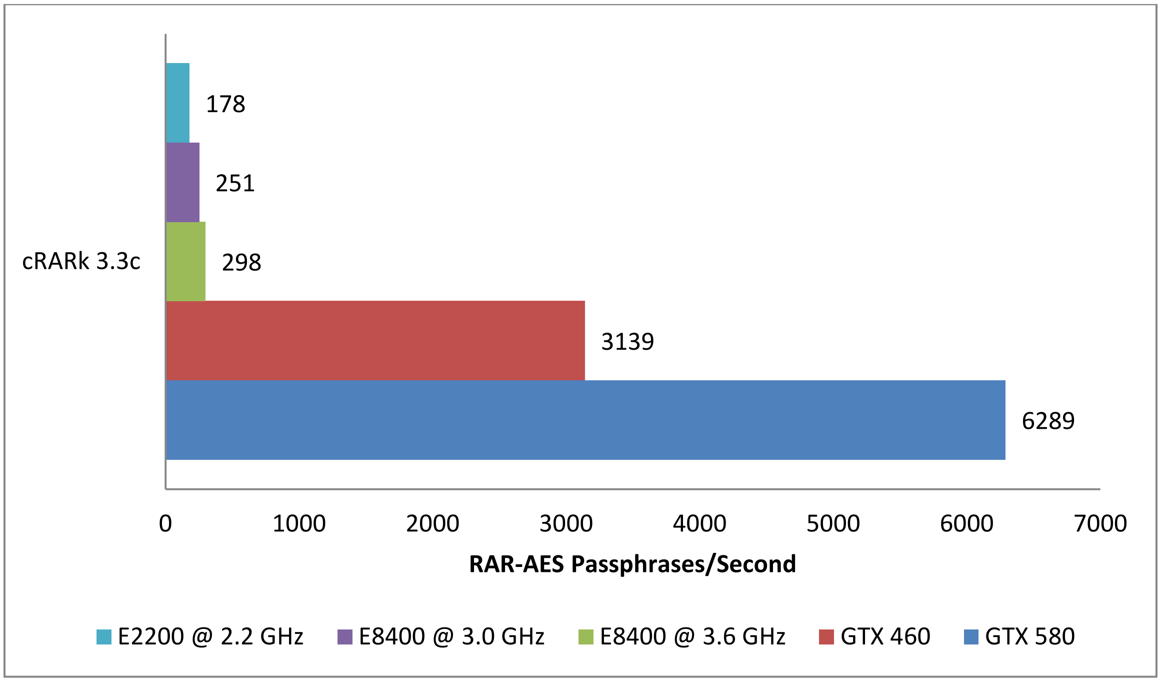

Figure 4 - RAR-AES Passphrases per Second by Device

Bibliography

Amazon Elastic Compute Cloud. (n.d.). Retrieved April 1, 2011, from Amazon Web Services: http://aws.amazon.com/ec2/

AMD Corporation. (n.d.). *An Introduction to OpenCL*. Retrieved April 13, 2011, from AMD: http://www.amd.com/us/products/technologies/stream-technology/opencl/pages/opencl-intro.aspx

Bailey, D. V., Batina, L., Bernstein, D. J., Birkner, P., Bos, J. W., Chen, H.-C., et al. (2009, November 5). *Breaking ECC2K-130.* Retrieved March 31, 2011, from http://binary.cr.yp.to/ecc2k130-20091105.pdf

Bernstein, D. J. (n.d.). The Factorization of RSA-1024. Chicago, Illinois.

Bernstein, D. J., Chen, T.-R., Cheng, C.-M., Lange, T., & Yang, B.-Y. (2008, November 3). *ECM on Graphics Cards.* Retrieved from http://eprint.iacr.org/2008/480.pdf

Bos, J. W., Kaihara, M. E., Kleinjung, T., Lenstra, A. K., & Montgomery, P. L. (2009, September 1). *On the Security of 1024-bit RSA and 160-bit Elliptic Curve Cryptography, V 2.1.* Retrieved April 2, 2011, from http://eprint.iacr.org/2009/389.pdf

Breaking ECC2K-130. (n.d.). Retrieved March 30, 2011, from http://www.ecc-challenge.info/

Cheung, H. (2005, April 7). *FBI Teaches Lesson In How To Break Into Wi-Fi Networks.* Retrieved March 27, 2011, from Information Week: http://www.informationweek.com/news/showArticle.jhtml?articleID=160502612

FIPS 197 - Announcing the Advanced Encryption Standard. (2001, November 26). *U.S. FIPS Publication 197*. National Institute of Standards and Technology.

Frankel, S., Eydt, B., Owens, L., & Scarfone, K. (2007, February). *Establishing Wireless Robust Security Networks.* Retrieved March 30, 2011, from National Institute of Standards and Technology: http://csrc.nist.gov/publications/nistpubs/800-97/SP800-97.pdf

Google Authenticator. (n.d.). Retrieved April 3, 2011, from Google Code: http://code.google.com/p/google-authenticator/

GPGPU. (n.d.). Retrieved March 23, 2011, from History of GPGPU: http://gpgpu.org/oldsite/data/history.shtml

IEEE Computer Society. (2007, June 12). Part 11: Wireless LAN Medium Access Control and Physical Layer Specifications. *IEEE Standard for Information Technology*. New York: NY.

Imhoff, R. (n.d.). Password Brute Force Calculator.

Kennedy, D. (2011, February 6). *Building the ultimate bad arse CUDA cracking server.* Retrieved April 1, 2011, from SecManiac: http://www.secmaniac.com/february-2011/building-the-ultimate-bad-arse-cuda-cracking-server/

Kleinjung, T., Aoki, K., Franke, J., Lenstra, A., Thome, E., Bos, J., et al. (2010, February 18). *Factorization of a 768-bit RSA Modulus.* Retrieved February 20, 2011, from http://eprint.iacr.org/2010/006.pdf

Kotadia, M. (2005, May 23). *Microsoft security guru: Jot down your passwords.* Retrieved April 2, 2011, from CNet News: http://news.cnet.com/Microsoft-security-guru-Jot-down-your-passwords/2100-7355_3-5716590.html

National Security Agency. (2009, January 15). *The Case for Elliptic Curve Cryptography.* Retrieved March 22, 2011, from National Security Agency: http://www.nsa.gov/business/programs/elliptic_curve.shtml

NSA Suite B Security. (2009, January 15). Retrieved March 11, 2011, from National Security Agency: http://www.nsa.gov/ia/programs/suiteb_cryptography/

NVidia Corporation. (n.d.). *What is CUDA?* Retrieved February 21, 2011, from NVidia Corporation: http://www.nvidia.com/object/what_is_cuda_new.html

Objectif Sécurité. (2010, February). Retrieved March 22, 2011, from https://www.objectif-securite.ch/en/news.php

Payment Card Industry Data Security Standard v2.0. (2010, October). Retrieved March 26, 2011, from PCI Security Standards Council: https://www.pcisecuritystandards.org/documents/pci_dss_v2.pdf

Payment Card Industry Data Security Standard v2.0 Glossary. (2010, October). Retrieved March 26, 2011, from PCI Security Standards Council: https://www.pcisecuritystandards.org/documents/pci_glossary_v20.pdf

Rivest, R., Shamir, A., & Adleman, L. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*.

Schneier, B. (2005, June 17). *Write Down Your Password.* Retrieved April 2, 2011, from Schneier on Security: http://www.schneier.com/blog/archives/2005/06/write_down_your.html

Schneier, B., & Ferguson, N. (2003). *Practical Cryptography.* Indianapolis: Wiley Publishing.

Sinnott-Armstrong, N. A., Greene, C. S., Cancare, F., & Moore, J. H. (2009, July 24). *Accelerating epistasis analysis in human genetics with consumer graphics hardware.* Retrieved March 20, 2011, from BMC Research notes: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2732631/

Tosaka. (2008, November 5). *CUDA Processing Flow.* Retrieved March 3, 2011, from Wikimedia Commons: http://commons.wikimedia.org/wiki/File:CUDA_processing_flow_(En).PNG

Vacca, J. R. (2009). *Computer and Information Security Handbook.* Burlington, MA: Morgan Kaufmann Publishers.

Academic Vita of Garrett M. Miller

Education: The Pennsylvania State University University Park, PA
Schreyer Honors College
B.S. with Honors in Information Sciences and Technology, Spring 2011
Minor: Security and Risk Analysis

Thesis: Future Implications of GPU Acceleration on Present Cryptographic Standards

Activities:

- Director, Undergraduate Learning Assistant Program, College of IST
- Alumni Relations Chair, IST THON
- President, IST Interest House
- Vice President, IST Student Government
- IST Honors Society: Gamma Tau Phi
- Treasurer/Seminar Leader, Information Assurance Club
- Seminar Leader, Security and Risk Analysis Club
- Defense Team Lead, IA Club iCTF Competition Team
- Member, Pride of the Lions Pep Band
- Learning Assistant, IST 297B: Supervised Experience in Instructional Support
- Teaching Intern, IST 110: Information, People, and Technology
- Teaching Intern, SRA 111: Principles of Information Security

Honors/Awards:

- Dean's List
- IST Undergraduate Research Grant
- Security and Global Scholars Program
- GEICO Achievement Award 2010
- IST Sophomore Student Leader of the Year 2008-2009
- IST Freshman Student Leader of the Year 2007-2008
- Carl and Patricia Henninger, Peter J. Lechner, Ms. Eva Blum, and Mr. Robert Bardusch Scholarships
- Pennsylvania Governor's School of Excellence for Information, Society, and Technology 2006

Presentations:

- Scanning, Probing, Penetrating (NMap, Nessus, Metasploit)
- Session Hijacking/Firesheep
- Spy Hunter Packet Challenge (Forensics)
- Fun with Wireless (802.11, Bluetooth)
- Network Poisoning/Eavesdropping
- Advanced Password Cracking