USE OF DIFFERENT GRAPHICS PROCESSING UNIT ARCHITECTURES TO ANALYZE VARIANCE IN HASH CRACKING RATE AND REAL WORLD IMPLICATIONS OF PASSWORD CREATION BY USERS

Miguel Luis G. De Joya
miguel.dejoya120399@my.lsgh.edu.ph

Niño Jose P. De Guzman
nino.deguzman@lsgh.edu.ph

Ma. Lilibeth E. Bilon
betching.bilon@lsgh.edu.ph

Alce M. Sentones
alce.sentones@lsgh.edu.ph

La Salle Green Hills, STEM Senior High School, Mandaluyong- Philippines

Abstract: The study looks at using different graphics processing Unit (GPU) architectures to analyze variance in hash cracking rate and real world implications of password creation by users. The study has two tests; the first experiment is coded with a benchmark setup where the two GPU Architectures are compared based on their hash cracking rates. The second proves which GPU Architecture performs better based on mean average using t – test of Independent Means. The third shows the implications of the raw results of the first experiment, it puts into practical use of real, randomly sampled hashes collected from a leaked database. Which is then decrypted using "hashcat". The first experiment's results are put through ANOVA to see if there is variance between GPU Architecture and Hash Cracking Rate. The Pascal and Maxwell architectures are used as a sample since they represent what a user would typically have in a system. However, the use different GPU Architectures show that there is no variance between the hash-cracking rates after being tested in ANOVA, which showed a P-Value of 0.084 at a significance level of 0.05. The second experiment proves that there is a significant difference between the Maxwell and Pascal Architecture, with Pascal having the higher average. Additionally, the recent GPU Architecture (Pascal) is utilized for its practical application of Hash Cracking Rate in five trials with a preset number of hashes, which proved to be intriguing due to the time taken for the randomly sampled hashes to get decrypted. The findings may be useful in the password creation and security of all individuals that use online services that require password creation.

Keywords: Graphics Processing Unit (GPU), hash cracking, hashcat, Pascal architecture, Maxwell architecture

Introduction

The use of passwords nowadays are common, anyone with some sort of Internet presence uses a password. Passwords are what keeps our online life, such as banking, email logins, computer logins, and more safe and secure. However, these can get compromised, it can be from password leaks, user ignorance, and password cracking/decrypting. Focusing on password cracking, specifically hashtypes, these are what plaintext passwords get encrypted to when they are registered by the user, however according to Kioon, et al., (2013), hastypes such as the Message Digest 5 (MD5) hash function, which is one of the most commonly used protocols, may contain security risks.

There are many types of attacks that can decrypt hash functions, the most common are the brute force attack, and dictionary attacks. With each method of attack has its strengths regarding efficiency and time taken, however there are also downsides. Based on a review done by Suchithra, et al., (2014), there is a set spectrum of possibilities regarding password cracking attacks. This review focuses on the information available to the attacker on the victim and the corresponding type of attack to use. Focusing one type of attack, brute force, it is the use of a multitude of potential alphanumeric combinations while having the ability of using non-dictionary words on the hash making it viable for more complex passwords.

Graphics Processing Units (GPU) was chosen instead of Central Processing Units (CPU) due to the fact that GPU's have many more cores. Supported by Nickolls and Kirk (n.d.), CPU's typically have an average of 4-8 cores with 8-16 threads (as of 2017). While GPU's can have hundreds of cores with thousands of threads, with more expensive models going further than that. With this information it can be inferred that GPU's may have better processing capabilities when faced with graphically intensive workloads, or workloads that require many cores (in this case has computing). Additionally, Graphics Processing Units are found and used in many computers nowadays, their most basic function is to render 2D and 3D graphics, images, and video that allow operating systems, graphical user interfaces, video games, and many more. The more advanced functions are used for highly multithreaded applications such as visual computing that provide real-time visual interaction with rendered objects through graphics, images, and video Nickolls and Kirk (n.d.). Graphics Processing Units are normally paired with Central Processing Units since the Central Processing Unit instructs the Graphics Processing Unit on the tasks that need to be done.

Numerous studies have been found on password cracking. Each using different methods, hardware and programs. Methods such as the brute force which tries every combination of hash, dictionary that use a predefined wordlist to compare against, and rainbow table attacks that use a space-time tradeoff algorithm that is similar to the dictionary attack. Hardware, in regards to using a different processing unit in this case using the Graphics Processing Unit (GPU) over the Central Processing Unit (CPU), since GPU's have more working cores, therefore allowing faster processing, according to Nickolls and Kirk (n.d.). Finally, the different use of hash cracking programs such as "hashcat", developed by Jens 'atom' Steube, "john the ripper", developed by Alexander Peslyak, and "cain and abel", developed by Massimiliano Montoro, all programs are open source, with each program having respective strengths and weaknesses are stated.

Useful information regarding the use of dictionary attacks can be found in (Olson, 2007) paper, which analyzed the simple ways to solve ciphers using a system called the dictionary attack, in which several optimizations were described as well as methods for effectively dealing with non dictionary words. It also reveals quantitative performance results for several variations of the approach, in addition to other implementations presented. However, the algorithm used in the experiment did not include non-English words and letters, and it was stated in the paper that it could be done provided it receives modifications.

In addition, (Weir, 2010) details new ways that information using probability can be used to maximize the success rate of hash cracking attacks. "From evaluating the usefulness of known techniques and models to creating unique techniques such as using probabilistic context others". In contrast, the paper does not cover the possibility of future ways of password creation, human nature, and future encryption technologies. While new types of prevention were created, there were also new ways to crack them.

In a similar study, (Sprengers, 2011) focuses on requirements regarding security and properties of password hashing techniques. Using a GPU to show a proof of concept that launches intensive processes to process data instead of a CPU to increase maximum hash rates. For future work, it would be able to intrigue, implement and optimize other password cracking methods, such as SHA-crypt, bcrypt, Windows NLTM, and Oracle's proprietary scheme. To add, it would be valuable to observe how Nvidia Graphics Processing Units perform compared to other chipsets manufactured by other companies.

All Processing Units (CPU, GPU, following), have what is called a core, in Central Processing Units these typically have an average of 4-8 cores. According to Nanehkaran and Ahmadi (2013), these are units that read and execute program instructions. Using common sense, more cores usually means more power, however it also means more energy is consumed. This is where threads are introduced, these threads are not physical, but they do increase the power of a processor by significantly reducing the time taken to process a command by queuing the next task beforehand. The benefits of threads range from video editing, 3D rendering and heavy multitasking.
This is supported by: Information regarding multi-core processing of Nanehkaran and Ahmadi (2013), which describes the trends of increasing speed a processor gets when more cores are added. It also shows the basic layout of a typical CPU, and shows the advantages of using certain layouts, advantages and disadvantages.

As regards to password cracking optimization methods, (Yiannis, 2013) describes the most popular types of password cracking techniques and creates optimized attacks using other techniques which optimize for performance such as the hybrid attack that utilizes rainbow table and wordlist algorithm, which used a leaked database (Phpbb) as a data set. However, at the time of data gathering, the paper's data set had not been 100% recovered (password database leak), therefore the whole list could not be used for the experiment since certain password were not recovered and multiple hashes are to be used for the experiment.

Additionally, (Kioon et al., 2013) analyzes the security strength of the Message Digest 5 (MD5) hashing algorithm and what happens if external algorithms such as salts and iterative hashing are used to further encrypt the hash. It states that the use of those external algorithms in the paper make it difficult for attackers to crack hashes as it has more characters in the hash. The researchers have also conducted an experiment with improved hashes using the external algorithms and most of time each attack or application has failed to crack the selected hash. The usage of the MD5 hashtype in this paper is towards decrypting it to show a practical application of password cracking in terms of real world implications.

In addition to that, (Suchithra et al., 2014) describes the different types/methods of how passwords are encrypted and stored onto servers. Also, he further stated the different ways to decrypt said passwords. Out of the varying types and methods stated in the paper, this paper focuses on one method, the brute force benchmark. This attack is used since the storage requirements for the files of the dictionary attack and the rainbow table attack are too large, ranging from Gigabytes to Terabytes.

Furthermore, the study conducted by (Kulkarni, 2015) and (Chester, 2015) examines how password cracking terms and modern applications that crack passwords. It also explains how implementations of two hash-based password-cracking algorithms are developed. This study as well as Kulkarni's talk about password cracking but this study specifies on applications and utilizes only one form of attack which was used for this paper as well.

In regards to multiple types of attacks for password cracking, (Kulkarni, 2015) discusses different methods of password attacks, various countermeasures for said password attacks, different authentication methods, and an analysis of different password attacks and their relative countermeasures. As the paper focuses on the brute force benchmark, this paper gives a more defined analysis on the said attack (other attacks as well) as well as their respective countermeasures.

With regards to password cracking and its possible effects to the society, (Hranicky´ et al., 2016) explains how information (passwords in this case) was leaked by "hackers". It also touches on how forensic experts legally decipher suspects' data as well as how they conduct password recovery.

Hong, (2016) emphasized on the "rainbow tradeoff" algorithm, auxiliary techniques were used to reduce the time taken for the algorithm to complete its goal. However, it requires delicate manipulations of the random function, therefore, making it a challenging task. While these techniques were not conducted due to the current timeframe and were not part of the experiment proper, this paper gave a better insight on how password recovery can be more efficient with these said techniques. Although it required careful manipulation and holds a large amount of risks, it showed how quick and efficient password recovery can be if done correctly. Another thing to consider, it does not use other forms of algorithms but does consider the possible outcomes with errors within the used algorithm.

Nickolls and Kirk (n.d.) focuses on GPU system architectures, its framework, and describes each parts functions. The paper also featured how each feature can be used to its best performance. Additionally, the paper defines the many uses for GPU's and it's architecture. It also introduces how Moore's law interacts with GPU's.

This paper is based on a research done by Chester (2015) on Analysis of Password Cracking Methods and Applications, which focuses on the various methods of password cracking and multiple applications for password cracking with varying results with the same number of hashes. Out of all the possible methods and applications found in the research, the paper focuses on the application "Hashcat" as it is the research tool being used. "Hashcat" is a commonly used program among hackers due to it being quick and efficient. In terms of its efficiency, it has a large amount of possible hashes that can crack and can do various attacks such as the Brute Force attack that are used for the paper.

The findings of this study aims to show the relative simplicity or complexity of how passwords get decrypted. It also aims to enlighten users on password strength and how to create them properly. With the rise of new security technology, users believe that their data is safe and they have nothing to fear. However, there is also an amount of ways to get through the vulnerabilities and steal data. Every individual with an online presence is impacted since they input their passwords day to day, with more than 80% of users using the same password with different variations according to a survey done by Dalieda (2017).

This research experiment determines the correlation of the use of different Graphics Processing Unit architectures to the rate of decrypting multiple hashtypes. There are many different methods of attacks and many different types of hash functioning. The first experiment utilizes the most common hashtypes, ranging from MD4 to ArubaOS. The next experiment only uses the MD5 hashtype, since it is the most commonly used hashtype, typically obtained from leaked websites or databases. In addition, for the third experiment, hashtypes with plaintext keyspaces over eight are coded to be automatically rejected, since keyspaces with nine and above take up too much time to decrypt. Additionally, ASIC cards are not used in the research since it is not on the "consumer grade" price. Finally, brute force is used for the third experiment, to simulate a scenario where the attacker does not have access to a large dictionary file for a dictionary attack, or a large rainbow table for the rainbow table attack.

The analysis of the experiment is done for only one day, and the data of experiment one has a sample of ninety-six hashtypes, applied to both the Pascal and the Maxwell GPU Architecture. However, some of the hashtypes for the first experiment may not be processed by the GPU because of unforeseen errors. On the other hand, the third experiment contains five trials based solely on the architecture with better results in the experiment to feature the real life implications of hash cracking using the brute force attack method found in hashcat. The second experiment does not take into account the practice of "salting", which is a process that farther encrypts a hash by providing a unique decryption key. Finally, all hashtypes found in the paper are commonly used hashtypes that originate from sites such as employee database passwords, blogspot logon prompts, and so on.

The researchers would like to seek answers on the following questions:

1. Do the two different GPU architectures (Pascal and Maxwell) process hashes in a way that one surpasses the other in terms of efficiency, regardless of raw performance?
2. How fast can passwords be decrypted using hashcat with either of the specified GPU Architectures (Pascal and Maxwell)?
3. How can users improve their password strength based on the results of this research?

Materials and Methods

This research follows a repeated-measures experimental design. It analyzed the time taken and number of successful passwords decrypted collected by the researchers in a random sample of a leaked database, which allowed the testing of password vulnerability. In more detail, this was a true experimental type of research since it was based on testing various samples. Additionally each item that was examined, in this case hashtypes, were tested in their "natural environment", which was the computer. The plan of the experiment was to observe the brute force attack on a random sample and analyze the time taken to decrypt the passwords. Lastly, the goal of this research was to spread knowledge on user password creation and security.

The experiment started with the researchers using the hashcat program and created a benchmark for both of the GPU's to analyze with ANOVA. Additionally, a random sample of five hundred hashes were subjected to brute force attack for five consecutive times, the time was recorded and analyzed. The control setup for the experiment was the i3-4150 Central Processing Unit to simulate hashrates with a consumer end system. Additionally, the treatment setup was the GTX 750ti and GTX 1050ti Graphics Processing Units, for the "enhanced" hashrates and password cracking times.

In this paper, two GPU's from different generations utilized (Pascal and Maxwell) to determine if the architectures of the said GPUs' process data differently (hashes). This paper acknowledges Moore's law (1965), however it does not take the law into account since efficiency of the architectures was being observed and not the difference in raw performance.

This study was based on analyzing the rate at which a hash is cracked and what GPU Architecture was more suited for cracking hashtypes. The study was conducted on a consumer grade computer using two different GPU's. This was done because the study required the researchers to collect and code the program needed to decrypt the password. Upon completion, the study was able to identify GPU Architecture that was most effective in terms of decrypting passwords.

Results and Discussion

The study was intended to know if either of the two Graphics Processing Unit Architectures processed hashes in a different mechanism due to the different ways that the Architectures are handled. As previously stated, a sample of ninety-six (96) hash-types, and three trials for both GPU Architectures used. On this sample, the experiment was conducted to find out if GPU's are changing the way that hashes are handled by GPU's for every new generation GPU Architecture. The results were then analyzed using ANOVA to determine if there was any observable difference when it comes to variance in efficiency.

After the first experiment, the researchers found out that the Pascal GPU architecture was superior to the Maxwell architecture. In addition, after the data was tested in ANOVA, there was no evidence that there was a variation in mean of the hashrates. Therefore, the Maxwell GPU Architecture has no advantages over the Pascal GPU Architecture. In addition, both GPU Architectures were not able to complete all benchmarks due to a system error that would not allow it to proceed. During the experiment, two hashtypes were unable to get analyzed due to the following error: "clGetEventProfilingInfo(): CL_OUT_OF_RESOURCES", this indicated that the GPU

Architectures cannot process such a hash.

For the third experiment, the data gathered showed promising results. Since the practical usage of a consumer grade GPU to crack hashes was relatively simple. It took only minutes to be able to crack five hundred randomly sampled hashes in one go. However, the researchers coded the brute force attack to reject hashes that ranged greater than eight keyspaces due to the results of a pilot study conducted prior to this research, which resulted in cracking rates that estimated in an exponential growth per added key-space. Therefore, even though GPU's can process hashes with relative ease but this does not count passwords that have keyspaces greater than eight.

Table 1. ANOVA computation (experiment 1)

Source of Variation	**Df**	**SS**	**MS**	**F**	**P-Value**
Treatments	5	47648817.66	9529763.532	1.9533	**0.084**
Error	558	2722434530	4878914.928		
Total	563	2770083348			

Table 1 shows the process of ANOVA at α=0.05 and the data that was put into the test. As can be seen, the P value of 0.084 is greater than α level of 0.05. It means that there is no significant difference among the means of the groups.
However, once the groups were analyzed, the experiment shows that there is no variance between the mean average of the two groups. This means that the two GPU Architectures used in the experiment perform at similar efficiency. Additionally, this may not be the case for other solutions for hash cracking such as ASIC cards as mentioned previously.

Table 2. t - test of Independent Means

	x	sd	df	t	p	Interpretation
Pascal	1658.708	2.568	562	3.136	0.00	Significant
Maxwell	1077.422	1.759		3.136		

At α=0.05, significant if p < 0.05

Table 2 shows that t-test of Independent Means between the Pascal and Maxwell groups. It shows that the mean of the Pascal group is greater than the Maxwell group. Additionally, the p value of 0.00 is less than α=0.05, therefore there is significant difference between the means of the Pascal and Maxwell groups.

Table 3. Time Taken for Pascal GPU to crack 500 Hashes (experiment 2)

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
Pascal GPU	0:28:19	0:19:21	0:25:03	0:27:26	0:25:01
In seconds	1699s	1161s	1503s	1646s	1501s

Many promising results can be seen from the results of table 3, as it reveals the speed at which hashes can be cracked. It shows that 500 MD5 hashes can be cracked within 30 minutes. This is an alarming result, considering the hardware used in the experiment can be easily acquired and allow individuals to decrypt passwords.

However, as mentioned previously, the passwords are only limited to eight keyspaces due to the exponential processing power required to proceed further. Therefore, it may be safer to use keyspaces that are greater than eight, due to the fact that it takes more time for individuals to decrypt the hash.

Conclusion

Password security is a widespread issue in current times. Every person with any kind of online presence uses one to protect his or her information, accounts, etc. Due to the use of poor passwords created by the users, hackers are able to exploit that vulnerability and are able to decrypt their passwords. This paper has shown the theoretical and practical side of password cracking, furthermore, its implications in the real world, with the decryption of the Message Digest 5 (MD5) hashtype. Results of the study concludes that the usage of different Graphics Processing Unit (GPU) Architectures does not have a varying difference from architecture to architecture, proving that the methods used by each Architecture to decrypt passwords is essentially the same. Elaborating further, it means that there is no specific consumer GPU Architecture that is specifically specialized for tasks such as password cracking. The results of the second experiment show how simple it is to crack hashes. Practical uses of this software paired with consumer grade GPU's can be used by hackers and potentially target a user with sensitive information, sniff their hashes, and cracking the hash within minutes.

The hypothesis of the study stated that the use of the Maxwell GPU Architecture has a hashrate mean that is more significant than the rest. The results of the study prove the hypothesis wrong with the use of one-way ANOVA. However, there are specialized graphics cards that are used by data centers and supercomputers that excel in processing large chunks of data at a time, which most of the time, hackers do not have therefore not a part of this research as mentioned previously.

Implications of this research are that newer GPU Architectures are able to decrypt passwords quicker, thereby allowing hackers to target a user, and withdraw their data, given that the user's password is within eight characters. Therefore this paper implies that users should use passwords than nine characters to circumvent a brute force attack against the user's password. However, simply using a long obvious password such as "password123" is not advisable, while it does stop the brute force attack but not the dictionary attack. Therefore, a recommended password should be a character which is over eight characters long and it should have random characters, an example would be "d#fd@g$4hs%!", which circumvents both the brute for and the dictionary attack. Finally, it is recommended to use different password for each account, so that when one account does get compromised, none of the other accounts get compromised.

In the future, studies should include testing for more GPU Architectures. In addition, the use of specialized ASIC cards as mentioned previously. This should be explored so that factors such as GPU Architectures outside of Pascal and Maxwell can be accounted to allow for more data accuracy.

References

Chester, J. A. (2015, Spring). Analysis of Password Cracking Methods & Applications. Retrieved August 30, 2017, from ideaexchange.uakron.edu/honors_research_projects/7/

Hong, J. (2016). Perfect rainbow tradeoff with checkpoints revisited. *PLoS One, 11*(11) doi:http://dx.doi.org/10.1371/journal.pone.0166404

Hranický, R., Holkovic, M., Matousek, P., & Rysavý, O. (2016). On Efficiency of Distributed Password Recovery. The Journal of Digital Forensics, Security and Law : JDFSL, 11(2), 79-95. Retrieved from https://search.proquest.com/docview/1825887998?accountid=28547

Kioon, M. C. A., Wang, Z. S., & Das, S. D. (2013). Security analysis of MD5 algorithm in password storage. Applied Mechanics and Materials, 347-350, 2706. doi:http://dx.doi.org/10.4028/www.scientific.net/AMM.347-350.2706

Kulkarni, S. K. (2015, November). A Survey of Password Attacks and Comparative Analysis on Methods for Secure Authentication. Retrieved August 30, 2017, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.390.2515&rep=rep1&type=pdf.

Nickolls, J., & Kirk, D. (n.d.). Graphics and Computing GPUs. Retrieved from http://www.comp.nus.edu.sg/~cs2100/2_resources/AppendixA_Graphics_and_Computing_GPUs.pdf

Nanehkaran, Y. A., & Ahmadi, S. B. (2013, June). The Challenges of Multi-Core Processor. Retrieved from http://www.ijoart.org/docs/The-Challenges-of-Multi-Core-Processor.pdf

Olson, E. (2012, February 10). Robust Dictionary Attack of Short Simple Substitution Ciphers. Retrieved August 30, 2017, from https://search.proquest.com/docview/213053404/fulltextPDF/AD1D4BD006B24E97PQ/1?accountid=28547

Sprengers, M. (2011, February). GPU-based Password Cracking. Retrieved August 31, 2017, from www.ru.nl/publish/pages/769526/thesis.pdf

Vijayan, Joy, J. P., & S. (2014). A Review on Password Cracking Strategies. Retrieved August 30, 2017, from www.ijrcct.org/index.php/ojs/article/view/664/pdf

Weir, C. M. (2010). *Using probabilistic techniques to aid in password cracking attacks* (Order No. 3442161). Available from ProQuest Dissertations & Theses A&I; ProQuest Dissertations & Theses Global. (849735356). Retrieved from https://search.proquest.com/docview/849735356?accountid=28547

Yiannis, C. (2013, May 01). Modern Password Cracking: A hands-on approach to creating an optimised and versatile attack. Retrieved August 30, 2017, from https://www.ma.rhul.ac.uk/static/techrep/2013/MA-2013-07.pdf