Hi, my name is Alex Kibis and I am here today to talk to you about an area of computing that is going through many changes today; cryptography.

Before I start however, is anyone here familiar with the term? (Look at class to see if anyone raises hand). Thank you for volunteering (state whether correct or incorrect; just say you were close if incorrect). Cryptography is the practice and study of securing communications in the presence of other parties. For example, if you send a letter to your parents, you expect it to get there without a hitch. What if the mailman is interested in reading what you wrote? You would be out of luck, since the majority of people don’t write in a secret language only understandable to them and their parents. That is where cryptography comes into play! (Click slide). When a packet of data is sent across the network, or stored somewhere on a drive, it is encrypted using a special algorithm which transforms the data into an unreadable string of characters. If the data is being sent somewhere, only the designated destination has the key to decrypt the jumbled data. Once this is done, the message is capable of being read again.

So how do these algorithms actually work? This slide shows a version of encryption called hashing, this is a one-way operation where the user will never be able to get back the original data. It is used to check the integrity of data (MD5 sums). It, along with other versions, are very parallel by nature. Before the message is actually sent, a key is used to take the data and perform several logical and elementary algebra operations, even permutations and combinations are used today. What would be a good way of taking advantage of this inherent parallelism?

With GPUs of course! With the inception of Nvidia’s 8000 series gpus, we have been able to program video cards to do more than just render frames on the monitor. AMD quickly followed up with their cards Radeon and FirePro series. Even users with built in video could take advantage of this extra processing power when intel released their HD2000/3000 video chipset. Finally, the newest addition to the club are Qualcomm’s Snapdragon processors which have been able to run native OpenCL code in Android version 4.4 KitKat and newer. How is encryption used today?

The cryptocurrency craze has taken hold of many of us. Encryption is used to secure user wallets, where coins are stored, in order to protect from unauthorized access. In addition, it is also used to protect the integrity of the block chains so they could not be corrupted in order to net more coins than designed. A more familiar system to most would be HTTPS, used by every online banking and online shopping site in the world. It features SSL encryption using AES to protect user data while doing their business online.

The newest cryptocurrencies are using the newest version of SHA hashing technology. These are MaxCoin, Copper Lark, and Slothcoin. While many are used for profit. Currencies such as slothcoin are meant to drive research further for digital currencies. What is SHA exactly?

It is a one-way hashing algorithm, remember what I said about data not being able to be retrieved? This is one of its implementations! The SHA hashing standard is managed by the NSA which requires strenuous security protocols in order to safeguard their data. The newest addition has been an algorithm called Keccak which is now used for SHA3. Keccak is in the sponge function family, meaning that the input and output size can be of any length which greatly improves the security of the data. Even the slightest change (1 bit) will change the hash entirely! How is this accomplished? With multiple stages of conditional operations.

The following is a snipped of code for the Keccak algorithm written in CUDA. It is one of many, I chose to show this particular version because it has full documentation and source for anyone interested. The thing to take note here is the immense increase of speed when using the GPU instead of the CPU for hashing. Testing showed an up to 80 times faster hash rate! That is 1GB/s vs mere MBs. What about SHA hashing on other systems?

I wanted to see the results of GPU implementation for myself instead of relying on other peoples results so I came up with a test on my own. For this test I used my personal desktop. It has the following specs. The CPU is a 4.5ghz Intel processor with up to 8 threads available for processing and high speed memory. The GPU is an older AMD 5000 series card with much lower clock speed but a great core count to make up for it. These tests were performed using the AIDA64 Extreme benchmark utility under Windows 8.1 64-bit. I should also mention that ram caching was enabled for this system (explain if people don’t know what it is).

The results were interesting and more surprising that I thought! Show strange results for memory speed. The algebraic operations were presumably faster on the GPU than the CPU. What I would like to focus on however is the SHA hashing. Please let me go to the next slides in order to more accurately show the differences between these tests. Here you can see the two groups of tests. The SHA performance is more than double on the GPU than what it is on the CPU! What makes this even more surprising is that the CPU is barely year old tech while the GPU is nearly 5 years old. Taking Moore’s Law into account, this would mean that a current-gen GPU would produce results which are 4 times better. It can be seen that this performance increase is likely attributed to the far superior integer performance of the card.

We’ve covered one-way encryption fairly well, what about data transfer? More specifically, wireless networking. It is in our homes, transmitting personal information. At work, pumped full with sensitive information. On our airplanes, for those on the go Facebook cravings. Even at Starbucks during our coffee breaks!

Many don’t realize it but wireless data can be sniffed fairly easily, using widely available network analysis tools such as WireShark and AirCrack. I’ve even tested this myself, for science of course. Some of the earlier versions of encryption such as WEP and WPA-TKIP were very ineffective at protecting data. WEP has a big overhead and takes under 5 minutes to crack using AirCrack. Others have similar fates. This is why AES is the most recommended encryption solution for wireless networks at the moment. It provides the most security, with least overhead. This is mainly due to AES having a hardware implementation with software-only for many others. It is used to encrypt the data stream flowing between the router and computer.

The proper term for two-way encryption is symmetric. AES specifically, is a standard set by the NSA and is used to safeguard government data, so you know it’s pretty secure. The protocol set up by them states that data designated Secret, requires AES 128, 192, or 256. Top Secret data is 192 and 256 only. It was only a matter of time before GPUs were used to accelerate this encryption protocol as well.

Unfortunately I was not able to find an AES benchmark that I could run myself on both the CPU and GPU. Thankfully AMD had my back on this since I found an entire white paper written by them detailing the performance improvements of running AES encryption on a GPU. It can be clearly seen from this graph that the GPU is much faster!

In fact the speedup factor is up to 16X!

Having shown that cryptography on GPUs is a venue that needs a good amount of attention, what’s next? GPU cryptography is young, but developing really fast. There are new bits of code available for individual testing coming out all the time. The true issue here is making sure it is very secure, and from what we’ve all seen, GPU code is much more complicated than its CPU counterpart. Thankfully several companies have come together to help improve upon this. The HSA Foundation’s goal is to change the way systems are designed in order to make it easier to program parallel devices. Just recently a new version of CUDA has come out which brought with it the unification of memory (memory is now global), drop in library support to add in parallel code libraries in place of serial equivalents, and higher GPU count per node has been achieved as well.

We are all moving towards what will prove to be an exciting few years in the cryptography industry! I’m glad to have been able to show you a small part of it. Thank you for your time. Any questions?