1. **The First Concept and Real-world Deployment of a GPU-based Thermal Covert Channel: Attack and Countermeasures**
   1. This article outlines the first documented GPU based Thermal Covert Channel attack.
      1. They achieve 8.75 bps with an error of less than 2%
      2. “In cyber-security, a covert channel attack is a means of communication between two applications or processes which are not allowed to communicate in a specific system [1].**”**
   2. **“**In a TCC, an attacker application (typically denoted as transmitter or malware) has infiltrated a secure zone (e.g., a Trusted Execution Environment or TEE). From there, the transmitter encodes sensitive binary data as temperature variations by performing intensive computing on its CPU when transmitting a binary 1 or leaving the CPU in an idle state when transmitting a 0. In a non-secure zone, a second malicious application (typically denoted as receiver or spy) reads the thermal sensors of its core, where the temperature variation produced by the transmitter is noticeable.”
   3. bc gpus have thousands of small cores with 1 thermal sensor for the entire device, you gotta do a shit ton of computations to get a noticeable temp change, but for short enough to let the device cool down quickly
   4. the initial way to tackle TCCs was to just do garbage computations to add extra noise
      1. seems wasteful and inefficient lulw
   5. typical attacks have 2 components, a transmitter in the TEE, and a receiver, in a non-secure zone
   6. **A close up of a text

      Description automatically generated**
   7. Countermeasures:
      1. GPU thermal Noise
      2. DVFS
         1. “A periodic DVFS policy that switches the frequency of the compromised CPU from the highest level to a lower level every channel period T.”
            1. Has a much greater performance loss on GPU than cpu
            2. Make it temperature aware lol
         2. In summary, our policy sets a target temperature defined as the base temperature under normal utilization. Then, while a covert channel is active, we read the current temperature value. If the temperature is higher than the target temperature (adding the sensor’s precision δ), we scale down the GPU frequency”. And vice versa
   8. **A close-up of a document

      Description automatically generated**
2. **Exploiting Parallel Memory Write Requests for Covert Channel Attacks in Integrated CPU-GPU Systems**
   1. A diagram of a computer system

      Description automatically generated
   2. We develop two attack variants that achieve a bandwidth of 1.65 kbps and 4.41 kbps and error rates of 0.49% and 4.32% respectively.
   3. Possible mitigations
      1. Prioritize memory read requests (use something better than drain\_when\_full)
      2. Partition channels in multi-channel MCs
3. **Hot Pixels: Frequency, Power, and Temperature Attacks on GPUs and Arm SoCs**
   1. 2 types of side channel attacks
      1. Physical side channels
      2. Microarchitectural attacks
   2. **A text on a page

      Description automatically generated**
   3. Put a shit ton of effects on pixels, examine the time it takes to apply those filters, determine the original picture
   4. **A black text on a white background

      Description automatically generated**
   5. **A table with numbers and symbols

      Description automatically generated**
   6. Mitigations
      1. Run systems well below power or thermal budgets so that frequency is never throttled.
      2. Isolate cookies from cross-origin iframes
         1. Already deployed in safari (possibly chrome?).
      3. Prevent SVG filters from applying to iframes or hyperlinks.
4. **Trident: A Hybrid Correlation-Collision GPU Cache Timing Attack for AES Key Recovery**
   1. To enable a full AES key recovery on modern GPUs, we propose Trident – a novel hybrid correlation-cache collision attack on GPUs. It is hybrid as the earlier key bytes (e.g., 1st-4th key bytes) are recovered using negative correlation while the latter key bytes (e.g., 5th-16th key bytes) are recovered using a cache-collision attack.
      1. 1st 4 bytes recovered if negative correlation is assumed
      2. Other 12 cannot?
   2. Attacks only carried out on unpopular cards
      1. Kepler, maxell, pascal, volta
   3. **A diagram of a computer program

      Description automatically generated**
   4. **A close up of a text

      Description automatically generated**
5. **GPU Side-Channel Attacks are Everywhere: A Survey**
   1. **A diagram of a computer memory

      Description automatically generated**
   2. “AES is a symmetric key encryption algorithm with a block of 128 bits and key of 128, 192, and 256 bits. It consists of 4 matrix operations: SubByte, ShiftRow, MixColumn, and AddRoundKey.
   3. ATTACKS LOOKED AT
      1. Coalescing Unit Attack
         1. “Recover last round key of Table-Based AES by exploiting coalescing Unit. Kernel execution time is linearly proportional to # of unique cache line requests that occur when accessing T(last)”.
         2. “By IDing stand out correlation coefficient in key bytes guess, we can recover all correct key bytes”.
      2. Internal Caches Attack
         1. On systems without a coalescing unit, the correlation between the number of unique cache line request by correct key guess and kernel execution time is very high.
         2. **A close-up of a memory card

            Description automatically generated**
      3. Memory Bank Attack
         1. **A text on a page

            Description automatically generated**
   4. All 3 of these attacks target a table-based AES.
   5. All 3 of these attacks successfully recover all last round key bytes in AES encryption.
   6. All 3 of these attacks exploit the GPU microarchitecture.
6. **A Novel GPU Overdrive Fault Attack**
   1. “Prior work has evaluated side-channel leakage on GPUs, including power consumption [4], electromagnetic emanation [5] and timing leakage of microarchitectural features (e.g., the GPU’s coalescing unit [6], [7], banked caches [8] and other timing-related features [9])”
   2. Overdriving is overclocking AND undervolting at the same time to violate this timing function
      1. Breaking this function results in random faults being gated in the output flip-flops, where the exact fault value is unknown and uncontrollable.
   3. **A close up of black text

      Description automatically generated**
   4. **A computer diagram with text

      Description automatically generated with medium confidence**
   5. **A diagram of a diagram

      Description automatically generated**
   6. 2 types of voltage frequency scaling
      1. Dynamic (DVFS)
         1. Has physical guard bands to accommodate for both environmental and process variations.
      2. Adaptive (AVFS)
         1. on-die hardware mechanisms adaptively choose lower-power higher-performing combinations of voltage and frequency by taking real-time measurements of the junction temperatures and voltages across different locations on the GPU. This method eliminates the power waste by removing the traditional guard band.
         2. entails significant reliability and security implications.
            1. This creates the vulnerability this attack exploits
   7. The attack can be easily prevented by having the kernel driver block any Operating Performance Points beyond discovered safe limits
   8. Change thresholds to lower than 2.5% or higher than 20%
      * 1. **A text on a page

           Description automatically generated**
7. **Hardware/Software Obfuscation against Timing Side-channel Attack on a GPU**
   1. AES encryption on a GPU compared to a CPU is a “double digit speedup”
   2. “Coalescing unit can dynamically merge multiple memory requests across different threads into a single memory access, called a transaction”
   3. this paper’s obfuscation method seems to just be throw a whole bunch of noise into the mix to make it harder to get data
      1. done thru randomizing the width of the coalescing unit across different cache lines
   4. “Side-channel attacks exploit information recorded during execution of encryption/decryption”
      1. “There are two main types of memory attacks: access-driven and time driven attacks. In an access-driven attack, the attacker exploits the principle of cache sharing to observe the victim’s memory access behavior. The adversary intentionally creates contention on the shared cache resource to be able to infer if certain addresses have been accessed by the victim. Alternatively, a time-driven attack is based on measuring the execution time of an encryption/decryption and establishing a correlation between this timing value and an embedded secret.”
   5. A close up of a text

      Description automatically generated
   6. **“**An ideal countermeasure increases the number of samples required for the attack, without hurting execution performance.”
   7. “In this paper we propose a GPU-oriented random rotating algorithm that works by rotating T-table addresses, changing the permutation over time. The frequency of applying rotation can be controlled to minimize the overhead, while providing the right level of obfuscation. Dynamic rotation also can exploit the parallelism of a GPU to prevent performance loss.”
   8. **A group of graphs with different colored lines

      Description automatically generated**
8. **Overdrive Fault Attacks on GPUs**
   1. **A diagram of a computer system

      Description automatically generated**
   2. **“**In this attack, fault injections are conducted through publicly available software interfaces to the GPU driver. These software interfaces allow users to inspect and/or to customize the GPU hardware settings, including voltage and frequency scaling (VFS), which is a common technique for power management on many computing platforms”
   3. Breaks the timing constraints of flip flops again lol
9. **Electrical-Level Attacks on CPUs, FPGAs, and GPUs: Survey and Implications in the Heterogeneous Era**
   1. **A diagram of a computer

      Description automatically generated with medium confidence**
   2. **“**a heterogeneous system includes a variety of computing platforms with numerous interconnect methods and integration levels between the components. With their chip-level integration, embedded system-on-chip (SoC) and multiprocessor system-on-chip (MPSoC) platforms are among the most notable examples of HCSs”
   3. 2 types of heterogeneous systems:
      1. Chip level integration
         1. Integration inside the device package
         2. CPU + real-time processing unit
         3. Components comm with the “advanced extensible interface”
         4. Access to main memory is often shared
         5. RowHammer exploits target systems where CPU, GPU, and FPGA can all access DRAM thru memory controller.
      2. System level integration
         1. typical of datacenters
         2. gpus and fpgas lie on their respective accelerator cards, connected to the CPU thru peripheral component interconnect express (PCIE), QPI, or UPI
         3. FPGAs and GPUs have access to the CPU’s main memory AND their own memory modules
   4. **RowA diagram of electrical attack

      Description automatically generated**
   5. **A diagram of a computer system

      Description automatically generated**
   6. RowHammer attacks induce errors in DRAM, affecting the entire system
   7. Side-channel analysis commonly uses
      1. Simple power analysis
      2. Differential power analysis
      3. Correlation power analysis
      4. Template attacks
      5. Mutual information analysis
      6. Machine learing
   8. Side channel attacks can target
      1. Hardware
         1. Power
         2. EM field
      2. Software
         1. CPUs usually have software that manages power consumption info
   9. Countermeasures include:
      1. Automated code modification
         1. Has performance overhead
      2. Lightweight cryptographic primitives
      3. Random code injection for obfuscation
   10. “Decreasing leakage from the hardware itself typically requires its redesign”
   11. FAULT INJECTION
       1. Hardware Based
          1. Voltage
          2. Frequency
          3. EM Field
       2. Software Based
          1. 5 generic steps of DVFS-based attacks
             1. Environment setup
             2. Profiling for an anchor
             3. Pre-fault delaying
             4. Fault injection
             5. Restoring the original voltage and frequency
          2. Frequency
          3. Voltage
   12. Implications for heterogenous systems
       1. **A table of research directions

          Description automatically generated**
10. **pacSCA: A Profiling-Assisted Correlation-based Side-Channel Attack on GPUs**
    1. “GPUs have recently become popular to host the encryption/decryption algorithms due to its high-throughput computing capability. However, the security issues of moving the cryptographic algorithms onto GPUs have not been studied adequately. Consequently, with absence of any protection strategy, the potential vulnerabilities of GPUs to side-channel attacks may expose the confidential information with high risk.”
    2. this article outlines an attack that rebuilds an AES-128 key in <6 seconds
    3. “Side-channel attacks have recently been approved to be feasible on GPUs for secret information leakage [1]–[6]. The research efforts in [1], [6] apply the correlation power analysis (CPA) on a GPU AES implementation, and succeed in extracting the secret key through analyzing the power consumption. By observing that the kernel execution time is linearly proportional to the number of unique cache line requests generated during a kernel execution which is dependent on the input data and encryption key, [5] successfully establishes relationship between the execution time of an encryption kernel and the encryption key. Based on the relationship, they then conduct a side-channel timing attack to recover all the 16 AES secret key bytes on a specific NVIDIA GPU [5]. [2], [3] implement side-channel timing attacks using similar scenarios with [5], however relying on different relationships describing the time of execution differentiated by the number of L1 cache bank conflicts [3] and shared memory bank conflicts [2] respectively”
    4. 3 types of AES encryption
       1. 128 bit with 10 rounds of encryption
       2. 196 bit with 12 rounds of encryption
       3. 256 bit with 14 rounds of encryption
    5. **A white text on a white background

       Description automatically generated**
    6. Most attacks seem to be just encrypting and decrypting a shit ton of files to reverse engineer the encryption key
    7. **A screenshot of a graph

       Description automatically generated**
11. **Side Channel Attacks in Computation Offloading Systems with GPU Virtualization**
    1. This paper looks at a scenario where the attacker and victim are in 2 different VMs, sharing the same GPU
       1. This concept extends to cloud computing based services?
    2. “a GPU typically works as an external PCI-E device. Therefore, the program does not directly execute on the GPU, instead, a runtime system schedules and offload tasks (program kernels) to the GPU device. This execution model makes the runtime library a potential target for attackers”
    3. **A text on a page

       Description automatically generated**
    4. VMs have separate virtual GPUs, but they still share the physical resources of the GPU is based on.
       1. This resource contention makes probing for AES keys easier?
    5. Run a resource intensive prober on the attacker system to slow the victim system.
       1. This works as a low res “performance counter”
       2. Use deep learning models on the gpu’s performance to inference what the victim is doing.
    6. Defenses
       1. Side-channel-aware Resource Scheduling
       2. Attacker Behavior Detection
          1. Have the hypervisor monitor GPU activities on each vGPU
12. **A Timing Side-Channel Attack on a Mobile GPU**
    1. “Mobile devices carry a significant amount of sensitive and personal data, including credit/banking transactions, medical records and passwords.”
       1. Target phones for PII and PHI
    2. This attack recovers an AES128 key by using known ciphertext and exploiting timing info on a mobile platform (ie Qualcomm Snapdragon)
       1. Apparently, this attack is generalizable to similar platforms
    3. “Because systems are often running multiple programs and processes, spy processes can infer a victim’s activities through characterizing the victim’s execution footprint on shared microarchitectural resource.”
    4. the mobile industry is utilizing GPU for both GPU and CPU functions lately
    5. timing based side channels on mobile platforms are noisy bc of thousands of concurrent threads and applications
    6. cache-based attacks have 3 types
       1. trace-driven
       2. access-driven
       3. time-driven
    7. One of the main concerns in performing timing attacks is the number of samples required. Timing attacks basically leverage the correlation between timing information and an intermediate value, depending on key (which is number of cache misses in this work).
       1. **A math equations and numbers

          Description automatically generated with medium confidence**
          1. T = time
          2. V = estimate of the intermediate value w/correct key byte
          3. P = correlation between time and intermediate value
          4. A = probability the max p > 0
          5. Z(a) = distance between distributions with p = 0 and max p
       2. **A graph of a graph with a red line

          Description automatically generated with medium confidence**
          1. this attack is successful with a correlation of less than 0.005 with ~ 1mil samples
13. **Cracking Randomized Coalescing Techniques with An Efficient Profiling-Based Side-Channel Attack to GPU**
    1. Randomized coalescing techniques are proven to improve security against side channel attacks
       1. fixed sized subwarp (FSS),
       2. random-sized subwarp (RSS),
       3. random threaded subwarp (RTS).
       4. “Kadam et al.’s research [14] shows that the RSS along with RTS outperforms other three coalescing randomization schemes (FSS, RSS, and FSS+RTS) on security enhancement and are demonstrated to enable 24 to 961 times improvement in the security against the correlation timing attacks with 5% to 28% performance degradation”
    2. The profiling-based side channel attack this paper proposes gets around this protection.
       1. The attacks assumes upper and lower bounds for the number of subwarps and can effectively ignore them.
    3. **A diagram of a computer network

       Description automatically generated**
    4. **A diagram of a computer

       Description automatically generated**
    5. **A diagram of a computer program

       Description automatically generated\**
    6. **A text on a white background

       Description automatically generated**
14. **Eavesdropping User Credentials via GPU Side Channels on Smartphones**
    1. “In this paper, we present a new side channel attack on mobile GPUs of Android smartphones, allowing an unprivileged attacker to eavesdrop the user’s credentials, such as login usernames and passwords, from their inputs through onscreen keyboard”
       1. “Our attack targets on Qualcomm Adreno GPUs and investigate the amount of GPU overdraw when rendering the popups of user’s key presses of inputs. Such GPU overdraw caused by each key press corresponds to unique variations of selected GPU performance counters, from which these key presses can be accurately inferred”
       2. >80% accuracy
       3. Negligible computing overhead and net traffic
    2. Defenses thru mitigations of access control on GPU performance counters or applying obfuscations on the values of GPU performance counters.
    3. “GPU performance counters (PCs) in certain categories reflect the amount of screen display changes at the granularity of individual pixels. This explicit and fine correlation allows direct eavesdropping without any ambiguity.”
    4. **A screenshot of a keyboard

       Description automatically generated**
    5. “We implemented our attack as an Android application running on the victim device, and evaluated the attack on Android smartphones with different Qualcomm Adreno GPUs.”
    6. The attack doesn’t need repetitive guesses, so inference latency is <0.1ms
    7. Mitigated by role-based access control
    8. GPUs have performance counters that measure overdraw in pixels
       1. Use these performance counters to rebuild passwords entered in keyboard GUIs
    9. **Diagram of a system

       Description automatically generated**
    10. Mitigations!
        1. Disable popups of key presses
        2. Malware detection
        3. Use access control in GPU PCs
15. **Can one hear the shape of a neural network? Snooping the GPU via Magnetic Side Channel**
    1. Audio based side channel attacks exist??
    2. “When concerned with machine learning, different network models exert different computational burdens on hardware [47]. The variance across operations and layers result in different physical patterns of consumption, regardless of implementation or hardware, leaving neural architectures in general susceptible to side channel attacks. SCA was recently used to infer machine learning models by observing power consumption profiles [9, 45, 47], timing information [11] and memory/cache access [19–21, 49]”
    3. **A diagram of a graph

       Description automatically generated with medium confidence**
    4. Neural networks have certain characteristics
       1. # of layers (depth)
       2. Sequence in which layers appear
       3. Each layer’s individual type (fully connected, convolutional, recurrent, pooling, etc)
    5. GPUs are used to accelerate a neural network’s step speed
    6. Different steps correspond to different electric loads and signal characteristics
       1. See the steps in C
       2. These steps correspond to different KINDS of operations, not specific values lol
       3. You can reverse engineer the mangnetic emanations to map a running neural net
    7. defensive measures
       1. prevent input sizes from being too large. Small batches dont emanate magnetic as much
       2. jam the signal by running separate kernels in parallel
16. **Power Analysis Attack of an AES GPU Implementation**
    1. Power leakage has 3 parts
       1. 2 linear
       2. 1 non-linear
    2. The article outlines how gpus work p well
    3. “Side-channel attack is a type of attack based on information gained from the physical implementation of a cryptosystem. Side-channel information can include power consumption, electromagnetic emanation, timing information, and even sound [8].”
    4. This paper uses Correlation Power Analysis (CPA)
       1. Correlation between observed power info generated by hardware and power estimation calculated from a power model (which is a function of the key).
       2. “For AES-128 with 16 bytes of key, the entire iteration would only be 2048 (= 28 × 16), much lower than the complexity of 2128 for a brute force attack.”
    5. Correct subkey values stick out after 40,000 traces
    6. Countermeasures
       1. Masking randomizes intermediate values
       2. Random delay of instructions does NOT work on GPUs
       3. Avoid using the same secret key for all the encryption on GPU.
          1. A couple of concurrent keys?
17. **Design and Implementation of Side Channel Attack Based on Deep Learning LSTM**
    1. “In paper [5], power consumption sidechannel attack idea was first presented in 1999.”
    2. Several SCA types, including
       1. Power Attack
       2. Electromagnetic Attack
       3. Timing attack
       4. Combined attacks
          1. Power consumption attack (aka energy analysis attack)
    3. “The steps involved in a CPA attack are summarized as follows:

• Create the power use trace matrix. If it is assumed that there are T traces, each with D data points, the power dissipation trace can be written as a matrix D of order T D.

• Produce a matrix for the range of midrange values. There are 256 different possible key combinations for each given byte, ranging from 0 to 255. The conclusion that can be derived from the plaintext byte that corresponds to it and the fictitious key used to build the value matrix V is that it is V T 256.

• Calculate the power matrix based on the presumptions. A matrix H with an order of T 256 is produced when the matrix V is mapped onto the Hamming weight model.

• Determine each column's Pearson correlation coefficient with respect to each column in matrix D. This calculation produces a matrix R with an order of 256 D. The size of the element rij increases with the degree of correlation between the jth column of matrix H and D. By finding the position of the element in the matrix R with the greatest value, it is possible to determine the location of the data point in the trace that is most related to the key according to its index. Finding the element's greatest value's location will help with this.”

* 1. Utilizing deep learning for side channel attacks requires satisfying all three of the following conditions:
     1. A close up of a document

        Description automatically generated
  2. The side channel attack based on deep learning is carried out in the manner described below:
  3. A close up of a text

     Description automatically generated