EMMasker: EM Obfuscation Against Website Fingerprinting

Mohammed Aldeen†, Sisheng Liang†, Zhenkai Zhang†, Linke Guo†, Zheng Song‡ and Long Cheng†
†Clemson University, SC, USA
‡University of Michigan – Dearborn, MI, USA
{mshujaa, sishenl, zhenkai, linkeg, lcheng2}@clemson.edu; zhesong@umich.edu

***Abstract*—Graphics processing units (GPUs) on modern computers are susceptible to electromagnetic (EM) side-channel attacks that can leak sensitive information without physical access to the target device. Website fingerprinting through these EM emanations poses a significant privacy threat, capable of revealing user activities from a distance. This paper introduces EMMasker, a novel software-based solution designed to mitigate such attacks by obfuscating the EM signals associated with web activity. EMMasker operates by generating rendering noise within the GPU using WebGL shaders, thereby disrupting the patterns of EM signals and confounding any attempts at identifying user online activities. Our approach strikes a balance between the effectiveness of obfuscation and system efficiency, ensuring minimal impact on GPU performance and user browsing experience. Our evaluation shows that EMMasker can significantly reduce the accuracy of state-of-the-art EM website fingerprinting attacks from average accuracy from 81.03% to 22.56%, without imposing a high resource overhead. Our results highlight the potential of EMMasker as a practical countermeasure against EM side-channel website fingerprinting attacks, enhancing privacy and security for web users.**

I. Introduction

Most modern computers leverage graphics processing units (GPUs) to enhance the performance of heavy multimedia and graphics tasks. The manufacturers of GPUs are expanding the capability of running multiple programs on GPUs for efficient parallel processing by providing a large number of streaming multiprocessors (SMs). However, several security concerns have been raised due to potential information leakage from the GPUs using side-channel attacks, including power consumption [19], timing delay [18], and electromagnetic emanations (EM) [5].

In particular, the EM emanations side-channel attacks have drawn the attention of many researchers in the security field [29]. Unlike other side-channel attacks, EM emanations attacks do not require physical access to the target device. Additionally, these attacks can be performed from a distance, making them a significant threat. EM signals often leak information that can be associated with specific activities and have been employed in the area of security attacks, such as cryptographic operations [2], displays [15], and current running programs [30]. Moreover, EM-based website fingerprinting attacks represent a sophisticated technique where an attacker can discern a user's web browsing activities by observing EM signals emanating from the GPU on the target machine. As web browsers' role has significantly expanded in the modern world, the sensitivity of the information they handle has also increased. As a result, web browsers have been the main focus of attacks designed to extract or seize control of users' sensitive data [36].

The variation of current in CMOS circuits is the root cause of EM emanations. Far-field EM emanations can propagate far or even through normal walls [41]. Although studies suggest using better physical EM shielding to prevent EM emanations and filter out currents and voltages [39], accomplishing such a goal at the level of power-hungry components like GPUs can be challenging, as metal-shielding does not completely eliminate EM signals [41]. Additionally, existing studies primarily focus on exploring alternative physical materials to enhance EM shielding capabilities [20]. There is a lack of research exploring software-based approaches to prevent website fingerprinting attack from EM emanations leakage.

In this work, we propose EMMasker as a practical software-based solution for mitigating website fingerprinting attacks associated with EM emanations. EMMasker is based on a simple yet effective approach that operates by emitting EM noise to disrupt the pattern of EM signals and confound any attempts at tracking or identifying user online activities through EM analysis. Furthermore, we deploy EMMasker as a browser extension that can be integrated into existing browsing frameworks, providing a practical and user-friendly solution for a wide range of users.

The previous research [41] has demonstrated that the electromagnetic emanations from the GPU memory clock can be exploited to mount long-range realistic attacks such as website fingerprinting attacks and keystroke attacks. The key idea in this work is that a GPU's workload fluctuates during the rendering of web pages, influencing the EM signal it emits. These workload fluctuations, along with the requirements to manage heat production, fan noise, and minimize power usage, result in alterations to the GPU performance levels, specifically, the P-states in NVIDIA GPUs. These

Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb) 2024
1 March 2024, San Diego, CA, USA
ISBN 979-8-9894372-2-1
https://dx.doi.org/10.14722/madweb.2024.23018
www.ndss-symposium.org

performance levels are dynamically controlled by the dynamic voltage frequency scaling (DVFS), resulting in corresponding adjustments to the GPU memory clock frequency [41]. This periodic clock signal is modulated by high-level computation activities unintentionally [3]. The vulnerabilities exploited in this study represent the core challenges that our proposed method strives to address and mitigate.

The main contributions of this work are summarized as follows:

- We present EMMasker, an innovative software-based mechanism designed to obfuscate the EM signals associated with web activity. By employing WebGL shaders to introduce rendering noise within the GPU, EMMasker disrupts the EM signal patterns, effectively confounding attempts at user activity identification. This approach represents a departure from traditional hardware-based solutions and contributes a new dimension to the field of EM side-channel defense.
- EMMasker is designed to balance the obfuscation of EM signals against the impact on system efficiency. The implementation ensures minimal impact on GPU performance and user browsing experience while providing significant obfuscation, confirming only a marginal increase of 7-13% in memory utilization.
- Our evaluation showcases the robustness of EMMasker against an advanced signal classifier. We illustrate how EMMasker can significantly reduce the accuracy of state-of-the-art EM website fingerprinting attacks on ResNet classifier, from average classification accuracy of 81.03% down to 22.56%.

II. Related Work

Various side-channel attacks for website fingerprinting have been introduced. Van Goethem et al. [34] explored how cross-site timing attacks utilize certain web features to obtain private information about a user, such as the user's activities on social networking sites. Kotcher et al. [14] proved that when a Cascading Style Sheets (CSS) filter is applied to the framed web page, its rendering time becomes proportional to the page information. Stone et al. [31] proposed redrawing events and measuring the time of applying the Scalable Vector Graphics (SVG) filter for history sniffing. The web browsers have decreased the timer resolution, which wipes out the timing signal utilized by these attacks. Oren et al. [27] used JavaScript to launch prime+probe attack on the last level cache. Another example is by Vila and Kopf [35], in which they proposed an attack on a shared event loop by enqueueing a number of brief tasks and observing the time at which these tasks were scheduled. The aforementioned side channel attacks were aimed at the operating system, browser, or CPU hardware level.

The electromagnetic emissions originating from the system components have been widely utilized in many side-channel attacks. TickTock [28] determines the microphone status of a laptop (on/off) based on the EM emissions generated from the wires and connections of the circuit conveying mic clock signals. In Periscope [12], the authors introduced a unique EM side channel attack from touchscreens paired with human activity to infer confidential inputs and recover 6-digit PINs. Also, the authors in [38] examined the camera (front/rear) operational status of a smart phone from the unintended EM emissions. A recent work [6] by Genkin et al. demonstrated that laptops' internal microphones can unintentionally capture the recording of computation-dependent electromagnetic EM leakage which makes EM-side channel attacks can be remotely launched.

Since GPUs emit relatively strong EM signal than other embedded processors, several attacks have been proposed to exploit GPU. Maia et al. [21] presented a framework that uses EM side-channel to obtain hyperparameters and network topology of a black-box neural network model. In [4], the authors examined EM leakage-based cache-collision attacks on AES implementations running on GPU and mount a key-recovery attack with Differential Electro-Magnetic Analysis (DEMA) utilizing extended simultaneous cache-collision in multi-threads systems. The authors in [5] leveraged an EM leakage side channel with a parallelism attack to exploit AES implementation against bitsliced GPUs, which are immune to cache-based attacks.

Zhan et al [41] proposed a practical attack on GPU utilizing EM leakage to disclose user online activity. Their model can intercept EM signal leakage at a distance up to 6 meters. This work has been proved to be feasible on current GPUs from various manufacturers. The authors reported their findings to AMD and Nvidia; surprisingly, neither companies have yet to find an immediately effective mitigation to this problem on off-the-shelf GPUs [41]. Specifically, in this paper, we propose EMMasker with the aim of defeating this attack.

III. Preliminaries

A. The Electromagnetic Side-Channel

Current flowing within circuits of the control, I/O, data processing, or other device parts creates EM emanations. Of these emanations, those induced by data processing operations carry the most promising information [1]. Ideally, the current only flows in the CMOS circuits when the logic state changes. And these state changes in data processing devices are controlled by square wave clocks. Each clock triggers a sequence of state-changing events which induce current changes in the data processing device. These current changes result in a sequence of EM emanations associated with state-changing events. These EM emanations carry information about current in circuits hence a sequence of states and events of the device. In other words, these EM emanations correlate with certain high-level activities. Recent research show that these emanations can be leveraged to launch realistic attacks [16], [7], [8], [40] as well as defences [11], [25], [42].

B. Dynamic Voltage and Frequency Scaling

Many modern processors cannot always run at their maximum clock frequency because of significant power consumption and enormous heat generated. The root cause is that

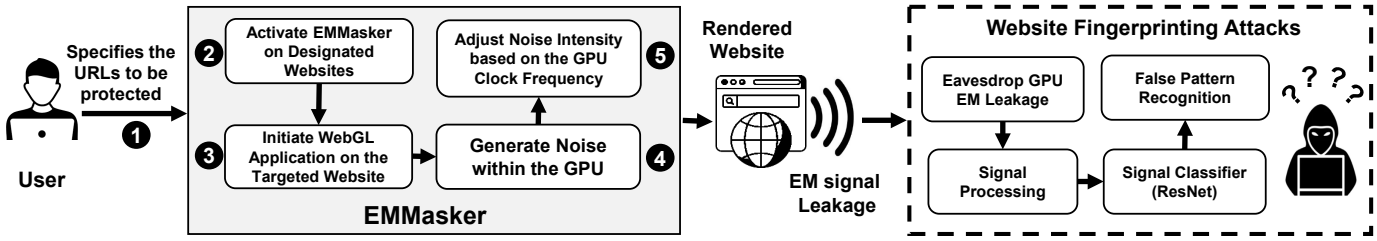

Fig. 1: EMMasker system overview

the dynamic power consumption P of a CMOS circuit is proportional to the clock frequency f and to the square of the voltage V ($P \propto fV^2$) [24]. Due to this relationship, modern processors keep their clock frequency and voltage as low as possible and only scale up when necessary. Dynamic Voltage and Frequency Scaling (DVFS) is the widely used technique in computing systems to achieve drastic power and energy savings [10] by matching the power consumption with required performance [37]. Specifically, DVFS changes both the voltage and frequency of the system with respect to the dynamic workloads. However, the clock frequency and the operation voltage are not unrelated. To make the processors function correctly, a higher clock frequency requires a higher voltage. Therefore, the clock frequency and processor voltage should not be regulated independently [24].

Nowadays, GPUs are becoming more and more powerful with the increasing demands of data-intensive applications. Meanwhile, power consumption has increased dramatically as well. For example, the Total Graphics Power (TGP) of the RTX 4090 GPU is up to 450 W, according to NVIDIA. It is almost doubled compared to RTX 2080, of which the TGP is about 225 W. To efficiently manage the power and reduce the amount of unnecessary heat generated, DVFS is deployed in modern GPUs to meet the required performance. With DVFS, recent modern GPUs define several power levels (the P-states defined by NVIDIA). For example, NVIDIA defines P-States ranging from P0 to P15, of which P0 is the highest performance/power state, and P15 is the lowest performance/power state or the idle state [26]. For a certain type of GPU, only a couple of states are available. These power levels are GPU active/executing performance capability and power consumption states [22].

IV. Design of EMMasker

A. Threat Model

EMMasker targets website fingerprinting attacks, where adversaries utilize EM field probes to monitor leaked EM emanations from the GPU and identify the running websites from a distance (far-field or near-field). With access to the captured EM emanations, they can apply signal processing techniques (time series derivation, noise contamination effect reduction) along with machine learning models such as the residual network (ResNet) [32] to predict the websites being accessed based on the signals that have been captured. We assume that the users will use EMMasker settings to select specific websites that they consider to be sensitive. As a result, EMMasker is triggered only when the designated websites are accessed. We also assume that EMMasker runs on a server (e.g., Flask) on the same machine to fetch GPU information for dynamic noise adjustments based on the GPU clock frequency in order to reduce GPU utilization.

It is worth noting that it is challenging to track the clock frequency in the full range due to the limited bandwidth of the radio device, such as software-defined radio (SDR). The compromised way for the attacker is to track a sub-range of whole clock frequency. For example, the research in [41] tracks clock frequency at the second lowest performance level for the attacks because they find that this frequency band is reached more frequently when rendering a web page. Therefore, we can leverage this characteristic to design a defense against this kind of EM side-channel attack. The attacker radio can not effectively receive useful information from the EM emanations if the clock frequency is out of the range of the designed frequency band of the attacker. To this end, one way is to force the GPU performance level to change dynamically by injecting some lightweight load to the GPU.

Our threat model assumes that the attacker is unable to monitor the entire spectrum including the full range of GPU frequency levels. Attackers could deploy multiple SDRs to cover a wider spectrum, or utilize advanced spectrum scanning techniques like SweepSense [9]. However, such advanced monitoring approaches require significant resources, making them less practical for most attackers. Our focus with EMMasker is to address common and more feasible attack scenarios.

B. Overview

Figure 1 illustrates the workflow of EMMasker within the web browser and the EM attack scenario from [41]. The user opens the browser and navigates to a website (❶). Upon visiting a website, EMMasker activates its jamming process automatically, as it is implemented as browser extension (❷). EMMasker initiates a WebGL canvas on the target website. This canvas is used for rendering noise directly within the GPU, whic is the main source of the EM emissions(❸). This can be achieved using the shader program within the WebGL to generate rendering noise and utilize for obfuscation purposes (❹). To ensure the patterns of the EM processed signals are fully obfuscated, EMMasker evaluates its generated noise dynamically based on GPU clock frequency (❺). It is

crucial to consider the potential impact on GPU resources. Therefore, EMMasker implements a delay to trigger jamming at specific times when the website is loading phase, effectively blocking the original generated EM signal.

C. Initiating WebGL Application

Upon installing and configuring EMMasker as a browser extension, EMMasker automatically activates whenever the user navigates to any website. EMMasker activates the jamming process when the user accesses one of the websites.

Given that the GPU is the primary source of EM emanations, we focus on generating noise within the GPU to directly obfuscate the emitted signal. Hence, we use WebGL (Web Graphics Library) [33] to generate noise exclusively within the GPU. WebGL is a JavaScript API that allows developers to access the GPU of a device to perform 3D and 2D graphics rendering in a web browser. However, some functions in WebGL are performed by the CPU, such as error handling, memory management, and input handling, while others are performed by the GPU, such as graphics rendering parallel processing and shader execution. It is worth noting that we bootstrap WebGL to only use the shader execution function, which is responsible for performing complex calculations. When the user accesses any of the designated websites, EMMasker acquires an HTML canvas element within the current website. This HTML canvas element acts as a container for JavaScript, which serves as a bridge between HTML and WebGL.

D. Generating Noise with Shader Program

To manipulate and render noise within the initialized WebGL canvas, we first need to create a shader program. This involves creating vertex and fragment shaders. Vertex shaders define the geometry and transformations, while fragment shaders define the appearance and colors of the pixels. We started by defining the objects to render. Within the vertex shader, we used displacement mapping to introduce noise to the rendering object. This is often done by perturbing the positions of the vertices based on noise functions or textures.

The fragment shader is then executed for each pixel on the rendering object and generates intensive rendering noise operations. These operations are designed to create a pronounced effect of rendering noise on the object. For instance, we used techniques that produce rapid variations in the object's appearance, such as shader-based animations, dynamic texture mapping, or procedural generation generates randomness based on factors, such as the fragment's position, a random value, or any other criteria.

E. Noise Rendering Management

It is crucial to ensure that the generated noise effectively conceals any pattern on the signal. Therefore, we first observed the inherent behavior of the GPU's graphics clock and memory clock frequencies, which can fluctuate based on the workload. Then, we monitor the GPU's clock frequency and adjust the noise intensity (*i.e.* on the fragment shader) in real-time. The noise amplitude is modified based on the current clock frequency, ensuring that the noise adapts to the changing of GPU clock frequency. For example, when the GPU clock frequency increases, EMMasker increases the noise intensity to ensure sufficient obfuscation and vice versa. Thus, even if an advanced classifier was applied. This dynamic noise method can effectively obfuscate the EM signal and ensure the adversary cannot find any evident pattern in the EM signal, even if deployed on advanced classifier.

Using multiple tabs simultaneously within the browser does not obfuscate the emitted signal. The reason is that popular browsers like Chrome and Firefox only send the workloads of the currently focused tab to the GPU for optimizing resource utilization [41]. To address this issue, EMMasker was implemented as a browser extension, which gained the ability to trigger its jamming mechanism whenever the user opens a new tab to navigate to another website within the same browser window. This approach ensures the noise generation is consistently applied across the defined websites opened in new tabs.

To minimize GPU resource utilization while maintaining effective signal obfuscation, it is crucial to ensure that the noise generated from EMMasker does not impact the GPU's performance. As a result, EMMasker triggers noise at random intervals every 1 to 15 seconds with each noise session lasting a brief 1 to 5 seconds. This approach allows EMMasker to strike a balance between signal obfuscation and GPU resource utilization, ensuring the noise is generated frequently enough to effectively obscure the signal, while also preventing excessive utilization of GPU resources.

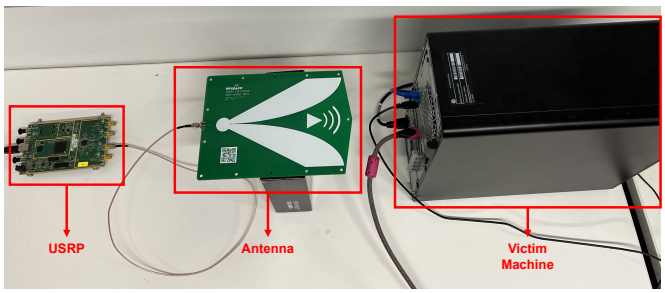

Fig. 2: Experimental setup: signal measurement equipment USRP B210

V. Evaluation

A. Implementation

We implemented EMMasker using the NVIDIA GeForce GTX 1650 (MSI) graphics card, which is built on the Turing architecture and equipped with 4 GB of GDDR5 memory. The experimental setup for EMMasker is depicted in Figure 2, which consists of a software-defined radio (SDR) device called USRP B210 and an ultra-wideband directional antenna. Furthermore, we utilized GNU Radio to manage the entire measurement process and handle the processing of the captured raw data. The SDR frequency was tuned to 810

MHz which is the second lowest frequency of the GPU and a 25 MHz sampling frequency was employed. EMMasker was implemented as a Chrome browser extension using JavaScript and was connected with a local server utilizing the Flask web framework enabling it to retrieve device information, such as GPU details. This connection allows EMMasker to dynamically adjust noise levels based on the frequency clock of the GPU.

Our evaluation is based on the attack proposed in [41], which allows the attackers to used EM field probes to capture EM emanations. This technique allows them to gather multiple traces from a victim's device, facilitating the development of a classifier adept at predicting user activities. The defense, EMMasker, is designed to obfuscate the EM signals associated with web activity, thereby eliminating discernible patterns. Thus, no matter how much training data an attacker collects, their classifier will not be able to identify any patterns, making the attack ineffective.

1) Capturing Signals: Capturing the emanated signal is the first step for adequately modeling the signal. Unfortunately, measuring the ideal emanated signal is not possible with only a one-time run of any website's signal is considered. One option is to collect many one-time run signals. The problem with this approach is capturing synchronized signals, i.e., the starting points of the captured website signals may differ from the exact start of a rendered website. To address this problem, we used a time synchronization protocol called network time protocol (NTP) [23] to provide accurate and consistent time across all devices on a network. Thus, we used NTP to compare the time on a client machine (to load the website) with the time on the reference server (to collect EM signals), then we matched the time reported by the reference server. Moreover, it automatically stops collecting signals as soon as the website is closed, avoiding additional signals from other activities that are irrelevant to website rendering. As a result, highly accurate emanated signals are collected without distortions from other activities, which could lead to false positives.

2) Spectrogram Samples: The EM signal is first transformed from a set of samples from a continuous stream into a vector, which can be processed as a block (window). Fast Fourier Transform (FFT) is then applied to each window to obtain its spectrum. In our measurements, the window size is determined by a named window function that assigns equal weights to all samples in the window. When performing spectral analysis using FFT, having a windowed signal will reduce spectral leakage and provide a balance between the clock frequency and time resolution. The rest of EM signal monitoring operates on this sequence of spectra, where each spectrum (i.e., the spectrum of one window) is referred to as a spectrogram.

3) Classification Model: We re-implemented the EM-based state-of-the-art side-channel attack [41], which served as the baseline to evaluate the effectiveness of EMMasker. We initially collected over 350 signals for each selected website (*i.e.*, Bing, Yahoo, Facebook, Google, and Amazon) using the same method described in Section V-A1. We proceeded to split the signals from each website into training and testing sets, consisting of 280 signals and 70 signals, respectively. For the classification task, employed the ResNet classifier (baseline) similar to the one utilized on [41], leveraging it for both the training and testing stages. This served as the control group, providing a baseline to evaluate the performance of EMMasker.

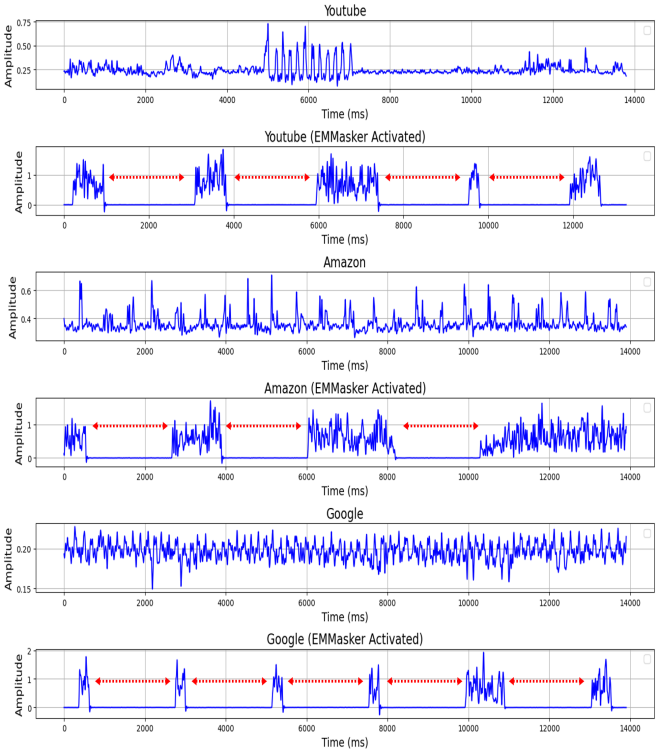

Fig. 3: Time series derived from the EM emanations that are measured when opening three websites using Chrome

Similarly, we then integrated EMMasker and re-collected the signals using the same methodology described earlier. It is important to note that we then trained the collected signals with EMMasker activated with ResNet classifier (EMMasker) to ensure a fair comparison between the baseline and EMMasker. Additionally, this approach allows us to evaluate how EMMasker can influence the classifier's ability to distinguish any patterns in the signals, thereby preventing website fingerprinting.

B. Evaluation Results

Figure 3 provides a visual representation of the EM signals for three selected websites as an example. Each website is represented by a pair of graphs: the first graph represents the original EM signal, and the second graph represents the EM signal after noise rendering from EMMasker. With EMMasker activated, the signals showed a dramatic change, becoming much more simplified with isolated peaks. From the figure we also notice that the signals amplitudes consistently remain at zero for a certain time, marked in red. This is owing to

EMMasker rendering noise shifted the GPU frequency clock beyond the 810MHz range, concealing any patterns in the EM signals.

ResNet classifiers were evaluated on the EM signals based on the baseline attack (ResNet classifier (baseline)) [41] as well as with the EMMasker (ResNet classifier (EMMasker)). In the absence of the EMMasker, it yielded an average website classification accuracy of 81.03% over the spectrum of websites tested. This performance benchmark reflects the attack's robustness in signal categorization in a noise-free environment. In contrary, EMMasker substantially diminishes classification accuracy to 22.56% across all websites, validating its design efficiency against EM website identification attacks.

In our evaluation, we examined the impact of EMMasker on GPU utilization. We monitored the GPU resources during the operation of EMMasker and compared the results with those of a standard baseline. From Figure 4, our findings indicate that EMMasker introduces a marginal increase in GPU utilization, with an average uptick of only 7 to 13 percent over the baseline. It's important to note that this percentage range is not constant as it reflects of a 1 - 5 second period of noise. Additionally, the GPU clock frequency shows a spike, suggesting that EMMasker triggers the GPU to operate at higher performance levels for brief periods. The memory usage, both general and GPU-specific, demonstrates a more variable pattern, yet stays within a moderate range of the baseline, which points to EMMasker's efficient memory management. Overall, the impact on resource utilization is moderate, indicating that EMMasker is a relatively resource-efficient tool when considering its noise rendering benefits.

Although we tuned the SDR to 810 MHz, in Figure 4 the GPU clock frequency was observed to be below 650 MHz. The GPU's clock frequencies are dynamically regulated by its power management system, a process known as dynamic frequency scaling. This system adjusts the GPU's performance by varying the clock frequencies in response to the computational demands, ensuring efficiency and reducing power consumption. In our experiment, for example, while the SDR was tuned to 810 MHz based on the assumption that this frequency is commonly used by the GPU for web page rendering, the actual measured frequency was lower, indicating that the GPU was operating in a lower power state during the experiment.

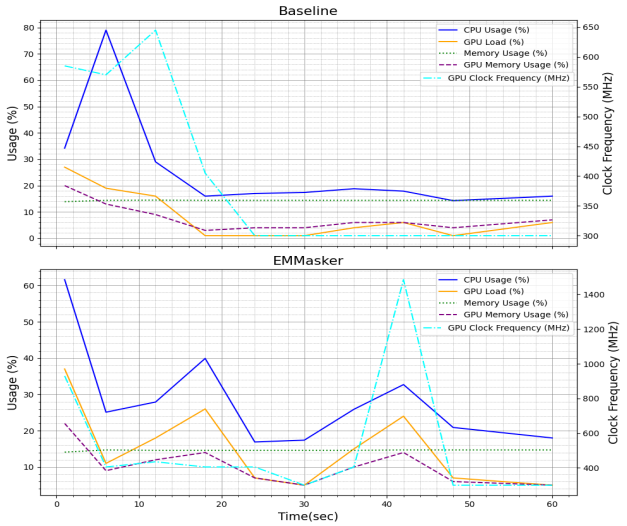

Fig. 4: Resources utilization overtime

VI. Discussion

A. Implications

While EMMasker primarily targets EM side-channel attacks, the noise generation technique utilized in this work can indirectly impact power consumption patterns, thus adding a layer of complexity against power analysis attacks. Power analysis attacks, like Differential Power Analysis (DPA) and Correlation Power Analysis (CPA)[17], rely on the correlation between power consumption patterns and cryptographic keys or other sensitive data.

Additionally, A study by Karimi et al. [13] demonstrated the feasibility of timing attacks exploiting EM emanations in mobile GPUs. EMMasker has the capability to introduce controlled noise into EM emissions, effectively masking the timing information that these attacks rely upon. By altering the EM signature, EMMasker could substantially reduce the accuracy of attacks attempting to deduce encryption keys or other sensitive data based on timing information.

B. Limitations

Despite the promising results, there are limitations to this study. First, EMMasker was evaluated on a subset of five websites, within a specific attack scenario, and on a specific GPU model. Expanding the evaluation to a larger set of websites could provide a broader perspective on its performance. Second, while our solution imposes minimal resource overhead, there is a marginal impact on GPU performance, which could potentially affect user experience on systems with less capable hardware. Third, the assumption that EMMasker will be hosted on a Flask web framework to function effectively may also not align with all end-user capabilities or preferences, potentially limiting its widespread adoption to some extent.

VII. Conclusion

EMMasker introduces a new defense mechanism against electromagnetic side-channel attacks on GPU-based web activities, striking a crucial balance between obfuscation effectiveness and system performance. It introduces noise and alters the signal characteristics leveraging WebGL shaders to generate dynamic noise within the GPU. Our evaluations confirm its efficacy in notably reducing attack accuracy, positioning it as a viable software-based solution for enhancing user privacy without necessitating substantial resource overhead or hardware alterations. Our contribution lays the groundwork for further exploration into software-based strategies, aiming to fortify the defenses against EM side-channel attacks and enhance security in our computing devices.

ACKNOWLEDGMENT

This work is supported by National Science Foundation (NSF) under the Grant No. 2239605, 2228616, 2147217 and 2114920.

REFERENCES

[1] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The em side-channel(s): Attacks and assessment methodologies," in *CHES*, vol. 2. Springer, 2002, pp. 29–45.

[2] M. Alam, H. A. Khan, M. Dey, N. Sinha, R. Callan, A. Zajic, and M. Prvulovic, "{One&Done}: A {Single-Decryption}{EM-Based} attack on {OpenSSL's}{Constant-Time} blinded {RSA}," in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 585–602.

[3] R. Callan, A. Zajić, and M. Prvulovic, "Fase: Finding amplitude-modulated side-channel emanations," *ACM SIGARCH Computer Architecture News*, vol. 43, no. 3S, pp. 592–603, 2015.

[4] Y. Gao, W. Cheng, H. Zhang, and Y. Zhou, "Cache-collision attacks on gpu-based aes implementation with electro-magnetic leakages," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 2018, pp. 300–306.

[5] Y. Gao, Y. Zhou, and W. Cheng, "Efficient electro-magnetic analysis of a gpu bitsliced aes implementation," *Cybersecurity*, vol. 3, no. 1, pp. 1–17, 2020.

[6] D. Genkin, N. Nissan, R. Schuster, and E. Tromer, "Lend me your ear: Passive remote physical side channels on PCs," in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022, pp. 4437–4454. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/genkin

[7] D. Genkin, L. Pachmanov, I. Pipman, and E. Tromer, "Stealing keys from pcs using a radio: Cheap electromagnetic attacks on windowed exponentiation," in *Cryptographic Hardware and Embedded Systems–CHES 2015: 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings 17*. Springer, 2015, pp. 207–228.

[8] ——, "Ecdh key-extraction via low-bandwidth electromagnetic attacks on pcs," in *Topics in Cryptology-CT-RSA 2016: The Cryptographers' Track at the RSA Conference 2016, San Francisco, CA, USA, February 29-March 4, 2016, Proceedings*. Springer, 2016, pp. 219–235.

[9] Y. Guddeti, R. Subbaraman, M. Khazraee, A. Schulman, and D. Bharadia, "{SweepSense}: Sensing 5 {GHz} in 5 milliseconds with low-cost radios," in *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, 2019, pp. 317–330.

[10] J. Guerreiro, A. Ilic, N. Roma, and P. Tomas, "Gpgpu power modeling for multi-domain voltage-frequency scaling," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2018, pp. 789–800.

[11] Y. Han, S. Etigowni, H. Liu, S. Zonouz, and A. Petropulu, "Watch me, but don't touch me! contactless control flow monitoring via electromagnetic emanations," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017, pp. 1095–1108.

[12] W. Jin, S. Murali, H. Zhu, and M. Li, "Periscope: A keystroke inference attack using human coupled electromagnetic emanations," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 700–714.

[13] E. Karimi, Z. H. Jiang, Y. Fei, and D. Kaeli, "A timing side-channel attack on a mobile gpu," in *2018 IEEE 36th International Conference on Computer Design (ICCD)*. IEEE, 2018, pp. 67–74.

[14] R. Kotcher, Y. Pei, P. Jumde, and C. Jackson, "Cross-origin pixel stealing: timing attacks using css filters," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 1055–1062.

[15] M. G. Kuhn, "Electromagnetic eavesdropping risks of flat-panel displays," in *Privacy Enhancing Technologies: 4th International Workshop, PET 2004, Toronto, Canada, May 26-28, 2004. Revised Selected Papers 4*. Springer, 2005, pp. 88–107.

[16] S. Liang, Z. Zhan, F. Yao, L. Cheng, and Z. Zhang, "Clairvoyance: Exploiting far-field em emanations of gpu to" see" your dnn models through obstacles at a distance," in *2022 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2022, pp. 312–322.

[17] O. Lo, W. J. Buchanan, and D. Carson, "Power analysis attacks on the aes-128 s-box using differential power analysis (dpa) and correlation power analysis (cpa)," *Journal of Cyber Security Technology*, vol. 1, no. 2, pp. 88–107, 2017.

[18] C. Luo, Y. Fei, and D. Kaeli, "Side-channel timing attack of rsa on a gpu," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 16, no. 3, pp. 1–18, 2019.

[19] C. Luo, Y. Fei, L. Zhang, A. A. Ding, P. Luo, S. Mukherjee, and D. Kaeli, "Power analysis attack of an aes gpu implementation," *Journal of Hardware and Systems Security*, vol. 2, pp. 69–82, 2018.

[20] H. Ma, M. Panoff, J. He, Y. Zhao, and Y. Jin, "Emsim: A fast layout level electromagnetic emanation simulation framework for high accuracy pre-silicon verification," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1365–1379, 2023.

[21] H. T. Maia, C. Xiao, D. Li, E. Grinspun, and C. Zheng, "Can one hear the shape of a neural network?: Snooping the gpu via magnetic side channel," *arXiv preprint arXiv:2109.07395*, 2021.

[22] X. Mei, Q. Wang, and X. Chu, "A survey and measurement study of gpu dvfs on energy conservation," *Digital Communications and Networks*, vol. 3, no. 2, pp. 89–100, 2017.

[23] D. Mills, J. Martin, J. Burbank, and W. Kasch, "Network time protocol version 4: Protocol and algorithms specification," Tech. Rep., 2010.

[24] K. Murdock, D. Oswald, F. D. Garcia, J. Van Bulck, D. Gruss, and F. Piessens, "Plundervolt: Software-based fault injection attacks against intel sgx," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 1466–1482.

[25] A. Nazari, N. Sehatbakhsh, M. Alam, A. Zajic, and M. Prvulovic, "Eddie: Em-based detection of deviations in program execution," in *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 2017, pp. 333–346.

[26] NVIDIA, "Nvapi reference documentation,gpu performance state interface," https://docs.nvidia.com/gameworks/content/gameworkslibrary/coresdk/nvapi/group__gpupstate.html, 2022, accessed: 02.08.2023.

[27] Y. Oren, V. P. Kemerlis, S. Sethumadhavan, and A. D. Keromytis, "The spy in the sandbox: Practical cache attacks in javascript and their implications," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1406–1418.

[28] S. Ramesh, G. S. Hadi, S. Yang, M. C. Chan, and J. Han, "Ticktock: Detecting microphone status in laptops leveraging electromagnetic leakage of clock signals," *arXiv preprint arXiv:2209.03197*, 2022.

[29] A. Sayakkara, N.-A. Le-Khac, and M. Scanlon, "A survey of electromagnetic side-channel attacks and discussion on their case-progressing potential for digital forensics," *Digital Investigation*, vol. 29, pp. 43–54, 2019.

[30] N. Sehatbakhsh, B. B. Yilmaz, A. Zajic, and M. Prvulovic, "A new side-channel vulnerability on modern computers by exploiting electromagnetic emanations from the power management unit," in *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2020, pp. 123–138.

[31] P. Stone, "Pixel perfect timing attacks with html5," *Context Information Security (White Paper)*, 2013.

[32] S. Targ, D. Almeida, and K. Lyman, "Resnet in resnet: Generalizing residual architectures," *arXiv preprint arXiv:1603.08029*, 2016.

[33] N. Trevett, "Webgl and webcl," *http://www. khronos. org/assets/uploads/developers/library/2012-pan-pacific-roadshow-March/WebGL-WebCL-China-Mar12. pdf, Zugriff am*, vol. 29, 2012.

[34] T. Van Goethem, W. Joosen, and N. Nikiforakis, "The clock is still ticking: Timing attacks in the modern web," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1382–1393.

[35] P. Vila and B. Köpf, "Loophole: Timing attacks on shared event loops in chrome," in *26th USENIX Security Symposium (USENIX Security 17)*, 2017, pp. 849–864.

[36] H. Wang, H. Sayadi, A. Sasan, P. S. Manoj, S. Rafatirad, and H. Homayoun, "Machine learning-assisted website fingerprinting attacks with side-channel information: A comprehensive analysis and characterization," in *2021 22nd International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2021, pp. 79–84.

[37] M. Wolf, "Chapter 5-processors and systems," *The Physics of Computing, M. Wolf, Ed*, pp. 149–203, 2017.

[38] B. B. Yilmaz, E. M. Ugurlu, M. Prvulovic, and A. Zajic, "Detecting cellphone camera status at distance by exploiting electromagnetic emanations," in *MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM)*. IEEE, 2019, pp. 1–6.

[39] A. Zajić and M. Prvulovic, "Experimental demonstration of electromagnetic information leakage from modern processor-memory systems," *IEEE Transactions on Electromagnetic Compatibility*, vol. 56, no. 4, pp. 885–893, 2014.

[40] Z. Zhan, Z. Zhang, and X. Koutsoukos, "Bitjabber: The world's fastest electromagnetic covert channel," in *2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2020, pp. 35–45.

[41] Z. Zhan, Z. Zhang, S. Liang, F. Yao, and X. Koutsoukos, "Graphics peeping unit: Exploiting em side-channel information of gpus to eavesdrop on your neighbors," in *2022 IEEE Symposium on Security and Privacy (SP)*, 2022.

[42] Z. Zhang, Z. Zhan, D. Balasubramanian, B. Li, P. Volgyesi, and X. Koutsoukos, "Leveraging em side-channel information to detect rowhammer attacks," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 729–746.