AuthEM: Authenticating and Monitoring ECUs Through Electromagnetic Signatures

Baki B. Yilmaz
Aether Argus Inc.
Atlanta, GA, US
baki@aetherargus.com

Zahir Khan
Aether Argus Inc.
Atlanta, GA, US
zahir@aetherargus.com

Elvan M. Ugurlu
Aether Argus Inc.
Atlanta, GA, US
elvan@aetherargus.com

Donald Greene
Aether Argus Inc.
Atlanta, GA, US
don@aetherargus.com

Angelos D. Keromytis
Aether Argus Inc.
Atlanta, GA, US
angelos@aetherargus.com

***Abstract*—This paper presents a non-invasive framework for monitoring and authenticating embedded systems using electromagnetic (EM) side-channel signals. Targeting automotive ECUs, the method compares EM emissions from test devices against golden references using STFT-based statistical features and KL divergence. The framework is validated on multiple commercial microcontrollers. It remains effective under interference and across different measurement setups. Results demonstrate improved detection accuracy over prior methods, highlighting the practicality and robustness of EM-based monitoring for real-world embedded system integrity verification, particularly in the context of automotive ECUs, where reliability and security are paramount.**

***Index Terms*—EM side-channels, malicious activity, hardware & software modifications, counterfeit, anomaly detection, ADAS.**

I. Introduction

Recent years have witnessed a dramatic shift in the design and operation of automobiles, evolving from isolated mechanical systems to sophisticated, connected, and increasingly autonomous cyber-physical systems. These advancements promise substantial benefits in terms of efficiency, safety, and user experience. However, this transformation has concurrently introduced significant security challenges. Modern vehicles are embedded with numerous communication interfaces, such as Bluetooth, Wi-Fi, and cellular, which expand their attack surface and make them vulnerable to a broad spectrum of cyber threats [1].

With the advancement of Advanced Driver Assistance Systems (ADAS), the attack surface for remotely compromising and controlling vehicles has further expanded [2]. Multiple high-profile incidents have demonstrated the real-world implications of vehicular cybersecurity weaknesses. Miller and Valasek successfully exploited vulnerabilities in the Chrysler Uconnect infotainment system to gain remote control over a Jeep Cherokee, impacting critical functions such as braking and steering [1], [3]. Similarly, security researchers demonstrated the exploitation of Tesla's browser to gain in-vehicle access [4]. Furthermore, government advisories have underscored the urgency for effective intrusion detection and response mechanisms in vehicular systems [5].

To address these security challenges, Intrusion Detection Systems (IDSs) have emerged as a vital defensive measure. IDSs can monitor both intra-vehicle networks (e.g., CAN, FlexRay, Ethernet) and inter-vehicle networks (e.g., VANET, IoV) for signs of malicious activity without disrupting normal operation. A wide range of IDS approaches have been proposed, including rule-based systems, traditional machine learning, deep learning, and hybrid methods, each with trade-offs in terms of accuracy, efficiency, and generalizability [1].

Despite significant progress, existing IDS methodologies often struggle to generalize across platforms or provide early detection in resource-constrained environments. In this study, we present a lightweight and non-invasive monitoring and authentication framework tailored for embedded automotive contexts, leveraging electromagnetic (EM) side-channel signals emitted from electronic control units (ECUs). The proposed framework is versatile and can be employed both at the acceptance stage, where ECUs are tested for counterfeit components or hardware modifications, and during runtime for continuous anomaly detection.

The organization of the paper is as follows: Section II provides background on EM side channels and details our signal processing methodology. Section III presents experimental results and discussion. Finally, Section IV concludes the paper with key findings.

II. Signal Processing & Comparison

As a step toward the broader goal of securing embedded systems in safety- and mission-critical domains, this work focuses on developing a non-invasive, zero-overhead framework for monitoring and authenticating ECUs based on their EM side-channel emissions. The core idea is to assess whether a

This work has been supported, in part, by DARPA contract 140D04-24-C-0026. The views and findings in this paper are those of the authors and do not necessarily reflect the views of DARPA. Distribution Statement 'A' (Approved for Public Release, Distribution Unlimited).

device under test (DuT) operates correctly by comparing its EM behavior to that of a trusted golden reference known to function as intended.

EM side channels are unintended byproducts of computational activity, arising from fluctuations in current flow within a device's circuitry, particularly due to transistor switching activity [6]. When a program is executed on a processor, the surrounding electric field changes systematically, and these variations directly correlate with the executed code. By monitoring such changes, one can gain valuable insights into the device's operational state and behavior. EM side-channel monitoring offers several advantages: it enables non-contact observation of internal processes, provides access to high-frequency bandwidths (including clock harmonics), and can therefore capture richer information. These properties make EM monitoring a dual use capability, serving either as a tool for adversaries seeking to exploit vulnerabilities or as a defensive technique to verify system integrity. Previous studies have used EM side channels to extract cryptographic keys [7], [8], profile memory access [9], detect malicious activity through neural networks [10], and identify hardware Trojans and counterfeits [11], [12]. In this work, we explore a non-adversarial application of EM side-channel analysis to ensure that ECUs behave as intended, detecting deviations that could otherwise lead to failures or even safety-critical consequences. Our approach is systematic and controlled: we execute known programs on test devices and monitor their EM emissions to extract and analyze behavioral patterns.

Among various side-channel modalities like power [13]–[15], we focus on EM signals due to several practical advantages. EM side channels offer high-bandwidth monitoring without requiring physical contact or consuming onboard resources such as memory or computation. Their fast response enables fine-grained temporal resolution, significantly outperforming slower side channels like temperature [16]. Unlike acoustic side channels [17], which suffer from low bandwidth and are easily masked by engine noise or vehicle vibrations, EM signals are more robust in noisy environments and better suited for concurrent multi-antenna monitoring, which can improve spatial resolution and support more advanced localization or source separation tasks. These properties make EM side-channel monitoring a uniquely viable choice for in-vehicle integrity verification.

The first step in side-channel analysis is to establish an appropriate experimental setup that maximizes the sensitivity of the monitoring system. While attackers may struggle to configure the system optimally due to access constraints, this limitation does not apply to defenders implementing a monitoring framework. Therefore, the placement of the probe can be optimized to enhance sensitivity, thereby extending the detection capabilities of the monitoring system. To this end, we execute a simple microbenchmark consisting of basic arithmetic operations within a for-loop, as described in [18]. This benchmark is known to generate peaks around the clock frequency and its harmonics, providing a predictable signal pattern for calibration.

Our investigation focuses on identifying the optimal probe location and center frequency that maximize the SNR of these peaks. We experiment with different sizes of Tekbox [19] and Aaronia [20] near-field probes to assess their performance. To measure signal power levels, we employ a Siglent spectrum analyzer (SA) [21]. Once the optimal center frequency and probe location are determined after a thorough investigation, we transition to using an Ettus B210 USRP SDR [22] to collect and sample the EM signals.

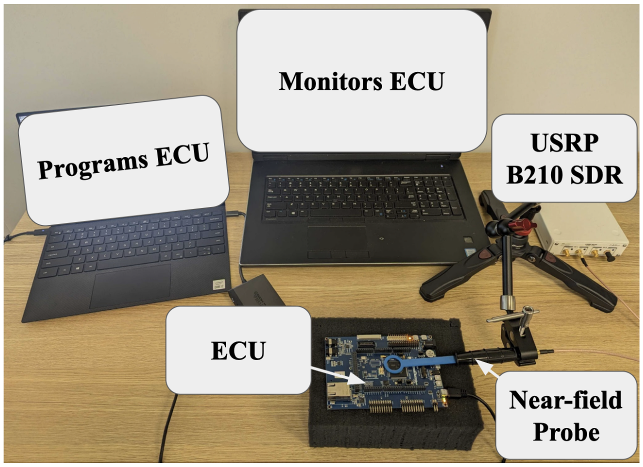


Fig. 1: Overall experimental setup for EM side-channel signal gathering.

An example of the experimental setup is shown in Fig. 1. In this configuration, one laptop is dedicated to programming the DuT, while another laptop runs our monitoring framework. It is important to note that the DuT and the monitoring laptop have no direct connection, and the DuT remains unaware of being monitored. Furthermore, there is no communication between the two laptops, ensuring that no information is transferred between them.

Once the EM signals are captured, the next step is to process them effectively. One of the primary challenges with EM signals emitted from devices is the phenomenon of signal smearing. This occurs due to imperfections in the clock frequency, aging effects, temperature fluctuations, and other factors, which cause deviations from the designated frequency [23]. As a result, the signal patterns may shrink, spread, or shift over frequency, complicating accurate analysis. Additionally, the high sampling rate required for capturing these signals generates large volumes of data, potentially introducing latency in processing and anomaly detection. In the context of vehicles and critical ECUs, such delays could lead to significant risks if detection does not occur promptly. To mitigate these challenges, we must reduce the data volume while preserving critical information within the signal. In this regard, we adopt the approach introduced in [24]. Specifically, we first generate a Short-Time Fourier Transform (STFT) matrix, where rows represent time intervals and columns correspond to frequency bins. Once the matrix is generated, we apply a max-pooling operation, with kernel sizes determined by parameters K_F and K_T, as illustrated in Fig. 2. This step effectively condenses the data by retaining the most prominent features in both time and frequency dimensions, thereby reducing processing latency while preserving key information for anomaly detection. This method also mitigates the smearing effect by condensing multiple frequency bins into a single bin, though at the expense of some resolution in both the frequency and time domains.

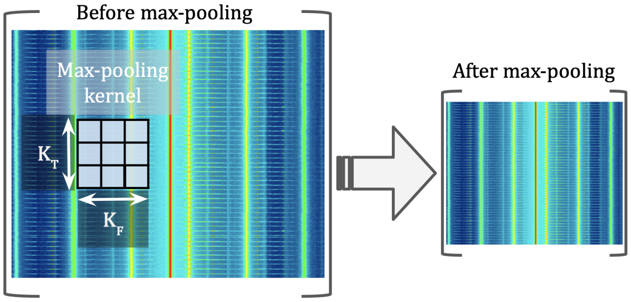
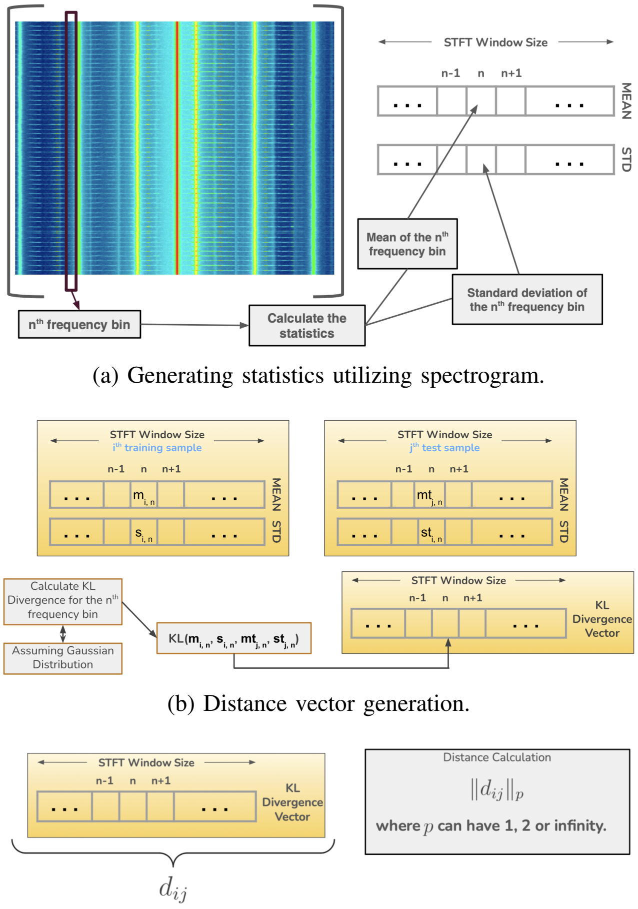


Fig. 2: Max-pooling to prevent signal smearing and data size reduction.

Once we obtain the reduced STFT matrix, the next step is to compare two signal snippets. An intuitive approach is to directly compare the STFT matrices, for example using the Frobenius norm. However, this method presents several challenges. One major issue is the substantial memory requirement needed to store large training datasets. These datasets are necessary because, in practice, the monitoring system may not start capturing data at exactly the same time as the reference (training) dataset with respect to program execution. This temporal misalignment introduces inconsistencies when comparing signal snippets, as the start and end points of the program's execution may not be synchronized between the training and test signals. Moreover, in real-world ECU operations, various functions are activated dynamically based on random events, making it practically impossible to collect sufficient training data that would ensure at least one sample perfectly aligns with every possible test signal. These challenges highlight the need for more sophisticated comparison methods that can accommodate temporal variability and dynamic system behavior.

To address this problem, the method introduced in [25] computes the mean of the STFT outputs (specifically, the mean of the rows of the STFT matrix) and compares these mean values instead of the entire matrix. This approach effectively reduces the data size for comparison (from millions of samples to a manageable STFT window size), thereby alleviating memory and computational demands. Additionally, it mitigates synchronization issues since ECUs often exhibit repetitive behavior, similar to an RTOS environment. By averaging the magnitudes of the STFT outputs, the approach accommodates temporal misalignments between training and test signals. However, this method has a significant limitation. It does not account for variations occurring at individual frequency bins across different STFT windows. For instance, if a frequency becomes activated during a particular STFT window, potentially signaling a malicious event, it may be obscured by the averaging operation. In this respect, we propose a methodology that considers both the mean and standard deviation of the frequency bins, as illustrated in Fig. 3.


STFT Window Size
n-1
n
n+1
...
MEAN
STD
Mean of the nth frequency bin
Standard deviation of the nth frequency bin
nth frequency bin
Calculate the statistics
STFT Window Size
ith training sample
$m_{i,n}$
$s_{i,n}$
jth test sample
$mt_{j,n}$
$st_{i,n}$
Calculate KL Divergence for the nth frequency bin
Assuming Gaussian Distribution
KL($m_{i,n}$, $s_{i,n}$, $mt_{j,n}$, $st_{j,n}$)
KL Divergence Vector
Distance Calculation
$\|d_{ij}\|_p$
where p can have 1, 2 or infinity.
d_{ij}

(a) Generating statistics utilizing spectrogram.

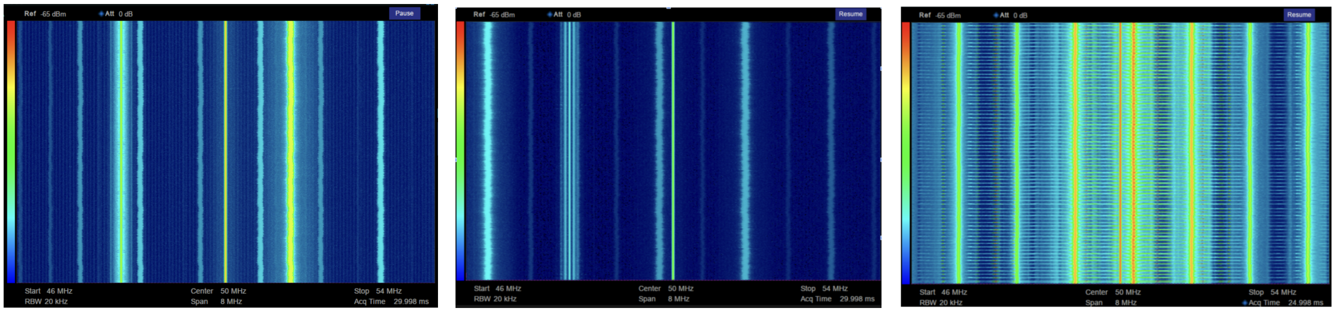
(b) Distance vector generation.

(c) Distance calculation.

Fig. 3: KL-Divergence based signal processing and comparison.

In Fig. 3a, we illustrate the computation of statistical features from a given STFT matrix, which is used to generate the spectrogram. Please note that this matrix is generated by using a single signal snippet. In this spectrogram, the vertical axis represents time, while the horizontal axis corresponds to frequency. Our methodology calculates statistics for each frequency bin. Specifically, the mean and standard deviation are computed for each bin, and these values populate corresponding vectors with sizes matching the horizontal dimension of the STFT matrix. When the max-pooling kernel size in the frequency domain is set to one, the sizes of these vectors are equal to the STFT window size. Finally, the training model is generated by collecting statistics from numerous signal snippets obtained while the device is engaged in legitimate activities.

Once the training signals have been collected and processed to gather these statistical profiles, the next challenge is determining how to compare the training signals with the test



(a) PIC32MZ2064DAR176.

(b) PIC32MZ2048EFM144.

(c) ATSAMV71Q21.

Fig. 4: Spectrograms obtained from different devices while running a similar program.

signals. In this work, we propose using the *Kullback-Leibler (KL) divergence*, under the assumption that the distributions of signal magnitudes in each frequency bin are approximately normal. This normality assumption greatly simplifies the calculation of KL divergence, as it can then be computed analytically between two normal distributions as follows [26]:

$$\mathrm{KL}(P \parallel Q) = \frac{1}{2}\left[\frac{(\mu_2 - \mu_1)^2}{\sigma_2^2} + \frac{\sigma_1^2}{\sigma_2^2} - \ln\left(\frac{\sigma_1^2}{\sigma_2^2}\right) - 1\right]$$

where $P = \mathcal{N}(\mu_1, \sigma_1^2)$ and $Q = \mathcal{N}(\mu_2, \sigma_2^2)$. This closed-form expression allows for efficient computation of divergence between the statistical profiles of training and test signals, facilitating real-time anomaly detection in monitoring systems.

By employing the KL divergence equation, we generate a vector of divergence values between the training signals and the test signal. An illustration of this process between the i^{th} training signal and the j^{th} test signal is provided in Fig. 3b. The resulting output is a multi-dimensional vector with a size equal to the number of frequency bins in the training or test signals.

Finally, as shown in Fig. 3c, we apply the p-norm to this divergence vector, where p can take different values. This parameter can be considered a hyperparameter of the proposed method, alongside others such as the STFT window size, max-pooling kernel dimensions, snippet time, sampling rate, and overlap rate. The use of p-norm enables flexible aggregation of frequency-wise divergences, allowing for tunable sensitivity to localized deviations in the signal.

III. Experimental Setup & Results

In this section, we present a series of experiments to demonstrate the effectiveness of the proposed model in identifying hardware modifications, including hardware Trojan injection, counterfeit detection, and general anomaly detection.

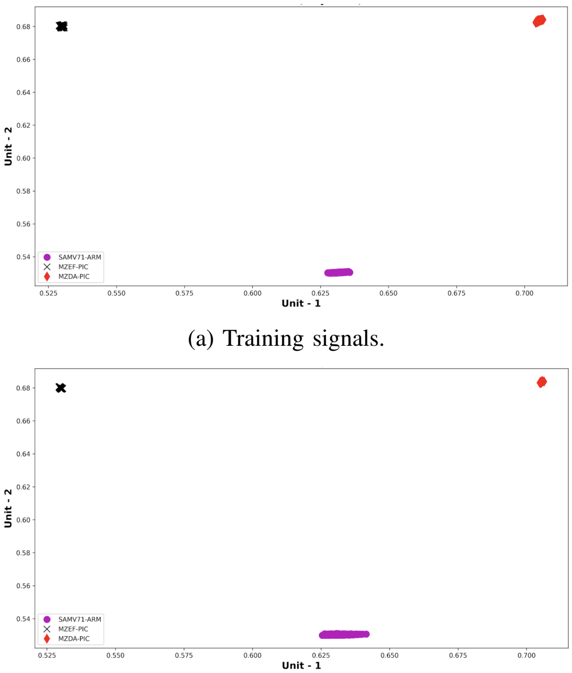
A. Hardware Modification & Counterfeit Detection

A common approach in the literature for evaluating the performance of hardware security frameworks is to use FPGAs, due to their flexibility in designing and programming different circuit configurations. However, in this study, we aim to validate our proposed framework on commercially available tape-out microcontrollers that are actually used in the automotive industry. One of the challenges we encountered is that vehicle manufacturers typically do not publicly disclose the exact microcontrollers used in their ECUs, nor do they provide accessible versions of them for research purposes, largely due to proprietary and confidentiality constraints.

To address this, we selected three microcontrollers that are publicly claimed to be widely used in automotive applications, according to vendor websites [27]. We refer to these devices as SAM [28], MZEF [29], and MZDA [30]. MZEF and MZDA belong to the same product family, and we therefore model MZDA as a hardware-modified variant of MZEF to simulate a modified hardware scenario, such as the presence of a hardware Trojan or unauthorized design change. SAM, on the other hand, is from a different microcontroller family, and we treat it as a counterfeit version of MZEF in our experiments. It is important to note that the selection of these microcontrollers is arbitrary and serves to illustrate the use-case scenarios of hardware modification and counterfeit detection.

For the experiments, we used the setup illustrated in Fig. 1 across all boards under evaluation. Each device was programmed to execute the same code sequence, designed to stimulate the compute unit as described in [31], enabling a fair comparison of their EM signal patterns. The corresponding spectrograms, collected using the SA at a center frequency of 50 MHz, are shown in Fig. 4. As anticipated, the EM patterns differ across devices, with MZEF and MZDA exhibiting higher similarity compared to SAM.

To further verify these differences, we applied the projection neural network method introduced in [24]. The underlying concept is that, if test signals are projected into the same feature space as their corresponding training signals, it implies the presence of consistent, device-specific signal characteristics. Conversely, overlapping projections across devices would suggest signal similarity. The input to the neural network is the mean vector of the processed signals, as described in Fig.3a. For this experiment, the STFT window size is set to 4096, the overlap rate between consecutive STFT operations is 0.1, and each signal snippet spans 50ms. The results are depicted in Fig. 5 showing that the training and testing signals from each device form clearly separable clusters. This confirms that the model is not overfitting and that the DuTs produce distinct EM signatures. Given that all devices were executing identical soft-



(a) Training signals.

(b) Test signals.

Fig. 5: Signal projections with a neural network [24].

ware, the observed differences in EM signals can be attributed to hardware-level variations. These findings demonstrate that the proposed framework is capable of identifying unauthorized hardware modifications and detecting counterfeit components.

B. Signal Interference Resilience

One common question regarding the proposed framework is its resilience to signal interference. In real-world scenarios, multiple components or devices often operate in close proximity, and a robust framework must account for the resulting interference to maintain high detection accuracy. To evaluate this, we designed the experimental setup shown in Fig. 6, where multiple boards are positioned as closely as possible to simulate a high-interference environment. We run the same program on each device and compare the EM signals captured in two scenarios: (1) when only the target device is powered on, and (2) when all devices are active simultaneously. The resulting spectrograms for the SAM board in both cases are shown in Fig.7.

A visual comparison of the two spectrograms reveals no significant differences, such as additional frequency activations or elevated background noise levels. This suggests that, due to the use of a near-field probe, the selection of an appropriate center frequency, and the inherently low-power nature of EM side-channel emissions, the proposed framework is largely robust to signal interference. However, to rigorously evaluate this observation, we conduct a one-class classification experiment.

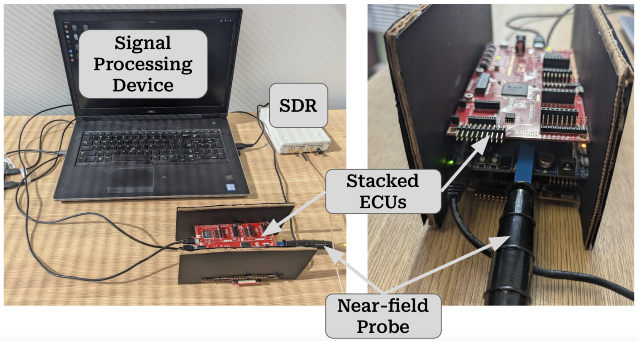
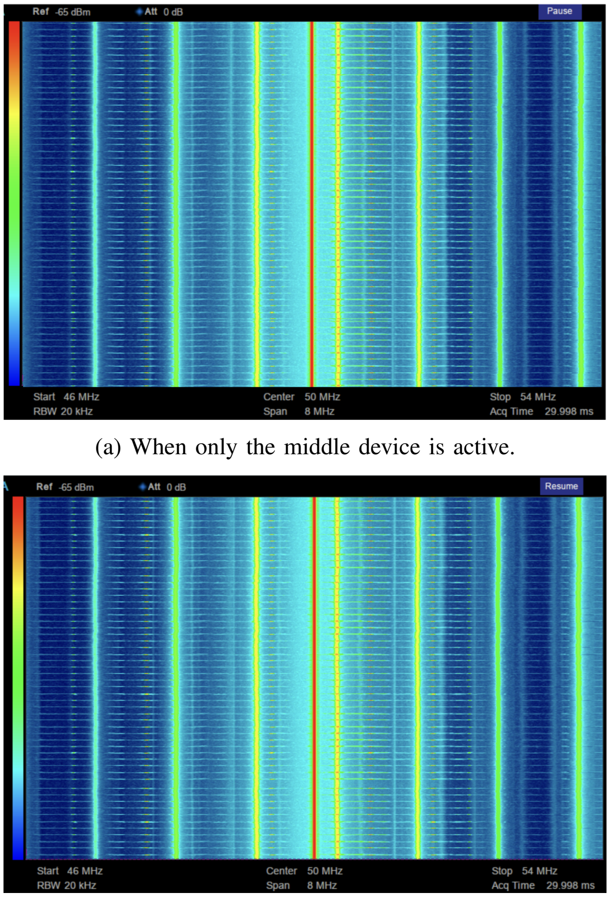


Fig. 6: Experimental setup for analyzing signal interference.



(a) When only the middle device is active.

(b) When all devices are active.

Fig. 7: Spectrograms for interference investigation.

Specifically, we train a model using the SAM board signals collected under the condition where only the SAM board is active, using the parameters described in the previous section. After the model is trained, we compute the distances between the test samples and the learned model for two conditions: (1) when only the SAM board is active, and (2) when all boards are active simultaneously utilizing the algorithm given in Fig. 3. If the distance distributions in both scenarios are similar, this suggests that the received signal patterns remain consistent despite the presence of nearby active components, indicating that the framework is robust to electromagnetic interference. Conversely, if the distributions differ significantly, it would imply that interference alters the EM signatures and potentially degrades detection accuracy.

The distance distributions of the test signals for the different

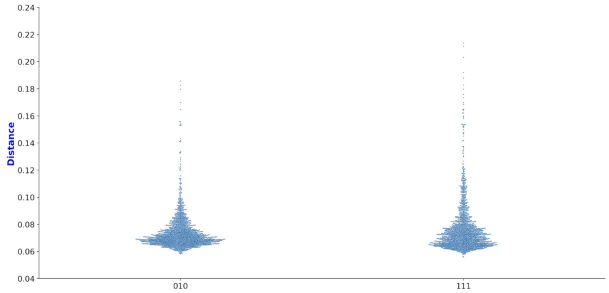


Fig. 8: Distances to training signals captured only when the middle device is active.

scenarios are shown in Fig. 8. In the figure, the binary labels (e.g., 010 and 111) indicate which boards are active, listed from bottom to top. For example, the distribution labeled 010 corresponds to the scenario where only the SAM board (the middle board) is active, identical to the condition under which the training signals were collected. In contrast, the distribution labeled 111 represents the scenario in which all three boards are active simultaneously, each running the same program. As shown in the figure, the distance distributions for both scenarios are closely aligned, supporting the conclusion that our framework is resilient to interference from neighboring devices.

C. Performance Comparison

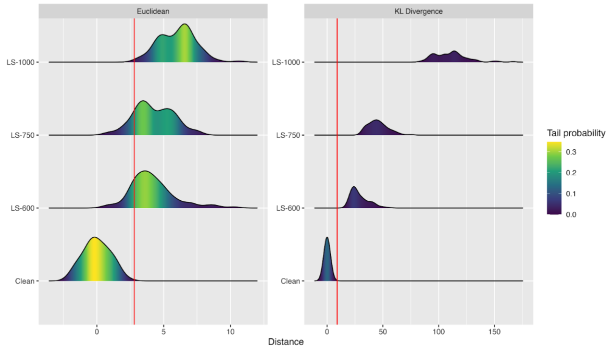
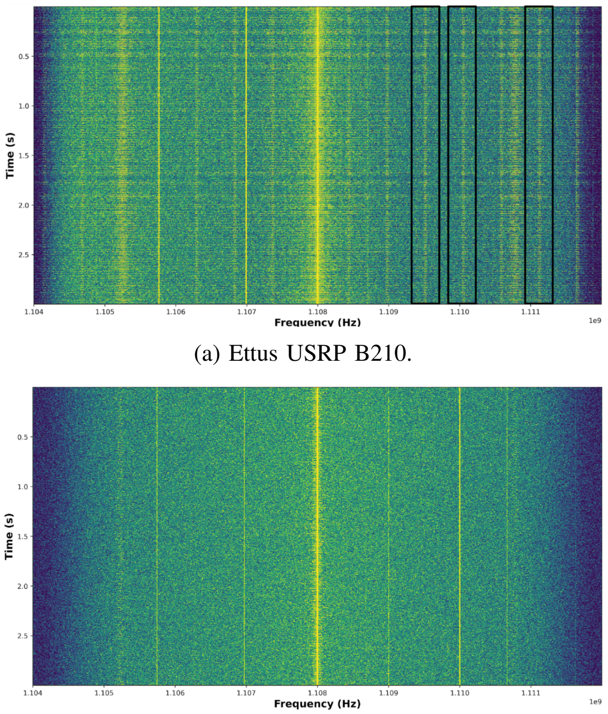


Fig. 9: Comparing algorithm performances.

To demonstrate that the proposed algorithm is also effective for malicious activity detection, we replicate the experiments presented in [24]. Specifically, we use the same experimental setup in which the system is compromised via a code injection attack. To evaluate the sensitivity of the detection framework, the injected payload consists of a simple for-loop, with the number of iterations varied to simulate different levels of attack intensity. By controlling the iteration count, we can assess the minimum detectable perturbation. We repeat the experiment exactly as described in the original work, and the results of our framework are shown in Fig. 9.

The results shown on the left side of Fig. 9 depict the distance distributions for different versions of the program using the method presented in [24]. In this figure, "benign" refers to the original, unmodified program, while the other labels correspond to code-injected versions, with the numerical values indicating the loop iteration count used in the injection. The corresponding results for our proposed framework are shown on the right. In both plots, the red line denotes the detection threshold. We observe that, even at the highest loop iteration count, the method from [24] struggles to distinguish the compromised versions from the benign one, resulting in a high false negative rate. In contrast, our proposed framework detects the injected code more reliably (even at smaller loop sizes) and exhibits an increasing separation between the threshold and distance values as the loop size grows. This widening gap is advantageous for reducing both false positive and false negative rates. These results demonstrate that the proposed framework is more sensitive and reliable in detecting malicious code injection activities.



(a) Ettus USRP B210.

(b) HackRF.

Fig. 10: Spectrograms obtained with different SDRs.

D. SDR Comparison

As the final experiment, we investigate the impact of measurement equipment on the performance of the proposed algorithm. To this end, we repeat the experiments described in Section III-C, this time using the HackRF SDR [32]. Unfortunately, the classification accuracy drops significantly, approaching 50%, which is comparable to random guessing. Further investigation revealed that the HackRF is limited in its ability to capture low-power EM emissions at the frequency range used in our setup.

The spectrograms obtained using both the HackRF and the Ettus B210 SDR are presented in Fig. 10. A key observation is that several active frequency components (typically appearing as vertical lines in the spectrogram) are not visible in the HackRF-captured signals. For clarity, we highlight some of these missing frequency components with boxes in the figure.

These results indicate that the choice of measurement equipment is a critical factor in ensuring reliable detection of malicious activity and unauthorized hardware modifications, regardless of the sophistication of the signal processing framework. Therefore, the entire pipeline—from equipment selection to signal processing—must be considered holistically to achieve accurate and dependable results.

IV. Conclusion

In this work, we introduced a side-channel-based hardware authentication and monitoring framework that utilizes EM side-channel signals to detect hardware modifications, counterfeit devices, and malicious code injections in embedded systems. By modeling spectral behavior through STFT-derived statistical features and applying KL divergence for signal comparison, the proposed approach achieves high accuracy in distinguishing between legitimate and compromised devices. We demonstrated its effectiveness on multiple commercially available microcontrollers using consistent program execution and validated its resilience against signal interference from neighboring devices. Moreover, our investigation into the impact of measurement hardware revealed that equipment sensitivity plays a critical role in maintaining detection performance, underscoring the need for a holistic design that includes both signal processing and hardware selection. Compared to prior work, our framework exhibits superior sensitivity and robustness. These results support the feasibility of deploying EM-based monitoring in automotive and other mission-critical embedded environments for proactive and passive security assurance.

References

[1] C. Lampe, R. Ernst, and F. Kargl, "Intrusion detection in the automotive domain: A comprehensive review," *ACM Computing Surveys*, vol. 55, no. 7, pp. 1–36, 2023.

[2] D. Nassi, R. Ben-Netanel, Y. Elovici, and B. Nassi, "Mobilbye: attacking adas with camera spoofing," *arXiv preprint arXiv:1906.09765*, 2019.

[3] A. Greenberg, "Hackers remotely kill a jeep on the highway—with me in it," https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/, 2015, accessed: 2025-06-04.

[4] SecurityWeek, "Pwn2own 2019: Researchers win tesla after hacking its browser," https://www.securityweek.com/pwn2own-2019-researchers-win-tesla-after-hacking-its-browser/, 2019, accessed: 2025-06-04.

[5] J. V. N. (JVN), "Jvnvu#99396686: Multiple vulnerabilities in automotive devices," https://jvn.jp/en/vu/JVNVU99396686/index.html, 2021, accessed: 2025-06-04.

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

[7] D. Genkin, L. Pachmanov, I. Pipman, and E. Tromer, "Stealing keys from pcs using a radio: Cheap electromagnetic attacks on windowed exponentiation," in *International workshop on cryptographic hardware and embedded systems*. Springer, 2015, pp. 207–228.

[8] 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 *Proceedings of the 27th USENIX Conference on Security Symposium*. USENIX Association, 2018, pp. 585–602.

[9] M. Dey, B. B. Yilmaz, M. Prvulovic, and A. Zajić, "Primer: Profiling interrupts using electromagnetic side-channel for embedded devices," *IEEE Transactions on Computers*, vol. 71, no. 8, pp. 1824–1838, 2021.

[10] B. B. Yilmaz, F. Werner, S. Y. Park, E. M. Ugurlu, E. Jorgensen, M. Prvulovic, and A. Zajić, "Marcnnet: A markovian convolutional neural network for malware detection and monitoring multi-core systems," *IEEE Transactions on Computers*, vol. 72, no. 4, pp. 1122–1135, 2022.

[11] X. T. Ngo, Z. Najm, S. Bhasin, S. Guilley, and J.-L. Danger, "Method taking into account process dispersion to detect hardware trojan horse by side-channel analysis," *Journal of Cryptographic Engineering*, vol. 6, no. 3, pp. 239–247, 2016.

[12] L. N. Nguyen, B. B. Yilmaz, C.-L. Cheng, M. Prvulovic, and A. Zajić, "A novel clustering technique using backscattering side channel for counterfeit ic detection," in *Cyber Sensing 2020*, vol. 11417. International Society for Optics and Photonics, 2020, p. 1141709.

[13] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Annual International Cryptology Conference*. Springer, 1999, pp. 388–397.

[14] Y. Liu, L. Wei, Z. Zhou, K. Zhang, W. Xu, and Q. Xu, "On code execution tracking via power side-channel," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 1019–1031.

[15] C. R. Aguayo González and J. H. Reed, "Power fingerprinting in sdr integrity assessment for security and regulatory compliance," *Analog Integrated Circuits and Signal Processing*, vol. 69, no. 2, pp. 307–327, 2011.

[16] M. Randolph and W. Diehl, "Power side-channel attack analysis: A review of 20 years of study for the layman," *Cryptography*, vol. 4, no. 2, p. 15, 2020.

[17] A. N. Alahmadi, S. U. Rehman, H. S. Alhazmi, D. G. Glynn, H. Shoaib, and P. Solé, "Cyber-security threats and side-channel attacks for digital agriculture," *Sensors*, vol. 22, no. 9, p. 3520, 2022.

[18] R. Callan, A. Zajic, and M. Prvulovic, "A Practical Methodology for Measuring the Side-Channel Signal Available to the Attacker for Instruction-Level Events," in *Proceedings of the 47th International Symposium on Microarchitecture (MICRO)*, 2014.

[19] TBPS01 EMC Near-Field Probes, https://www.tekbox.com/product/tekbox-tbps01-emc-near-field-probes/.

[20] AARONIA PBS, https://www.tequipment.net/Aaronia/PBS1-5/Standard/Passive-Oscilloscope-Probes/?rrec=true.

[21] Siglent SSA3000X-R Real-Time Spectrum Analyzer, https://siglentna.com/spectrum-analyzers/ssa3000x-r/.

[22] Ettus USRP B210 SDR, https://www.ettus.com/all-products/ub210-kit/.

[23] B. B. Yilmaz, N. Sehatbakhsh, A. Zajić, and M. Prvulovic, "Communication model and capacity limits of covert channels created by software activities," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1891–1904, 2019.

[24] D. Greene, Z. Khan, A. D. Keromytis, and B. B. Yilmaz, "Heterogeneous ic component identification via em side-channels," in *MILCOM 2024-2024 IEEE Military Communications Conference (MILCOM)*. IEEE, 2024, pp. 1058–1063.

[25] B. B. Yilmaz, E. M. Ugurlu, F. Werner, M. Prvulovic, and A. Zajic, "Program profiling based on markov models and em emanations," in *Cyber Sensing 2020*, vol. 11417. SPIE, 2020, pp. 69–83.

[26] Y. Zhang, J. Pan, L. K. Li, W. Liu, Z. Chen, X. Liu, and J. Wang, "On the properties of kullback-leibler divergence between multivariate gaussian distributions," *Advances in Neural Information Processing Systems*, vol. 36, pp. 58 152–58 165, 2023.

[27] Microchip, "32-bit mcus for automotive," https://www.microchip.com/en-us/solutions/automotive-and-transportation/automotive-products/microcontrollers/32-bit-mcus.

[28] ——, "Atsamv71q21," https://www.microchip.com/en-us/product/atsamv71q21.

[29] ——, "Pic32mz2048efm144," https://www.microchip.com/en-us/product/pic32mz2048efm144.

[30] ——, "Pic32mz2064dar176," https://www.microchipdirect.com/product/PIC32MZ2064DAR176-I/2J?productLoaded=true.

[31] F. T. Werner, B. B. Yilmaz, M. Prvulovic, and A. Zajić, "Leveraging em side-channels for recognizing components on a motherboard," *IEEE Transactions on Electromagnetic Compatibility*, vol. 63, no. 2, pp. 502–515, 2020.

[32] HackRF One: Software Defined Radio, https://greatscottgadgets.com/hackrf/.