Electromagnetic Insights Acquisition Through a Forensics-as-a-Service Platform

Senal Punsara
University of Colombo
School of Computing
Colombo, Sri Lanka
kksenalpunsara@gmail.com

Dinil Ratnayake
University of Colombo
School of Computing
Colombo, Sri Lanka
dinilsratnayake@gmail.com

Janitha Devin Ratnayake
University of Colombo
School of Computing
Colombo, Sri Lanka
janithadevin@gmail.com

Asanka Sayakkara
University of Colombo
School of Computing
Colombo, Sri Lanka
asa@ucsc.cmb.ac.lk

Akila Wickramasekara
University College Dublin
Dublin, Ireland
akila.wickramasekara@ucdconnect.ie

***Abstract*—Electromagnetic Side-Channel Analysis (EM-SCA) has been demonstrated as a viable technique for extracting forensic insights from Internet of Things (IoT) devices. However, the complexity of the data acquisition and analysis processes demands a high level of technical expertise from forensic investigators, which poses practical challenges in real-world scenarios. To address these challenges, this research proposes a Forensics-as-a-Service (FaaS) platform aimed at mitigating technical barriers by streamlining the EM-SCA process. This research introduces EMvidence, a FaaS platform designed to simplify and automate Electromagnetic (EM) data acquisition, handling, and analysis in IoT forensic investigations. The platform comprises two components: the EMvidence data acquiring application for efficient data capture at crime scenes, and the EMvidence web application for cloud-based analysis of EM data. Comprehensive evaluations were conducted on key aspects, including data acquisition, file transfer, pre-processing, and analysis, to assess the platform's effectiveness in supporting forensic investigations.**

***Index Terms*—Digital Forensics, Electromagnetic Side-Channel Analysis (EM-SCA), Forensics-as-a-Service (FaaS)**

I. Introduction

The digital forensics field encompasses the identification, acquisition, preservation, and analysis of evidence from various computing and communication devices [1]. While traditional methods, such as file system forensics, are effective for desktops and laptops, accessing non-volatile memory in mobile devices often requires risky and invasive approaches that can compromise data integrity and violate forensic soundness of the procedure [2]. IoT devices, storing data primarily in volatile memory, present additional challenges due to their heterogeneity and lack of standardized interfaces. Invasive techniques, such as chip-off forensics [3], [4], are impractical and time-consuming in most cases [5]. Therefore, the need for non-invasive methods, such as EM-SCA, is evident [6]. EM-SCA allows for data acquisition without physical tampering of the target device, offering a viable solution for IoT device forensics without compromising device integrity.

The application of EM-SCA in IoT forensics holds significant potential for deriving forensic insights. However, its direct adoption is hindered by the absence of specialized tools and the high level of technical expertise required from investigators. While previous research has demonstrated the effectiveness of EM-SCA in detecting software behavior and cryptographic algorithms with high accuracy [7], Current methods require knowledge of digital signal processing, and digital forensic investigators must manually conduct the process using the GNU Radio library [8] and Software Defined Radio (SDR) devices [9]. Additionally, the computational demands of analyzing EM data, particularly when integrating Machine Learning (ML), further complicate its implementation on standard personal computers, emphasizing the need for a practical, cloud-based forensic platform to bridge these gaps.

This work introduces EMvidence, an open-source FaaS platform designed for EM-SCA. The platform facilitates EM data acquisition through the EMvidence data acquiring application, while the EMvidence web application enables users to upload, analyze EM data, and generate detailed forensic reports. It also allows the forensic community to integrate custom analysis modules, providing flexible insights into a wide range of IoT devices under investigation.

The rest of this paper is organized as follows. Section II discusses the related work on EM-SCA and other related technical implementations. Section III provides the design and implementation of EMvidence data acquiring and web applications. Section IV contains the results and evaluation of this EMvidence framework. Finally, Section V concludes the paper by highlighting the future work directions.

II. Related Work

In the realm of EM-SCA, there has been a diverse set of methods utilized to acquire EM data, from the usage of probes as discussed by Gandolfi et al. [10] to the introduction of

979-8-3315-1142-5/24/$31.00 © 2024 IEEE

SDR devices by Nguyen et al. [11] and providing advanced forensic insights by Sayakkara et al. [12] from the EM data acquired using SDR devices. While these approaches offered valuable insights, they also underscore the need for specialized software tools tailored for EM data acquisition. Similarly, existing pre-processing techniques in the EM-SCA domain show a consistent sequence of steps across studies, indicating potential for standardization and efficiency gains. However, the lack of dedicated forensic tools for pre-processing EM files and seamlessly integrating them into the analysis workflow poses a notable challenge. Developing such forensic tools could significantly enhance the practical adoption of EM-SCA in digital forensics investigations, thereby unlocking its full potential for probing IoT devices and other applications.

A FaaS platform provides digital forensic services over the cloud, enabling users to access forensic tools and resources remotely. The review of FaaS platforms highlights the importance of comparing and contrasting different FaaS models to aid decision-making for researchers and practitioners. While challenges such as processing large data volumes are acknowledged, a deeper exploration of emerging trends and security considerations is needed. Miller et al. emphasize the importance of encryption for ensuring data confidentiality and integrity in a FaaS model. However, integrating hashing algorithms alongside encryption can further enhance data integrity during the transmission of data from the local computer to the cloud in a Faas platform [13].

EM data tend to be extremely large in size, and therefore, requires efficient handling while ensuring security and integrity. Mallafi et al. explore file compression techniques using Asynchronous JavaScript And XML (AJAX) and Web-service technologies, demonstrating improvements in upload efficiency [14]. Sayakkara et al. delve into the Hierarchical Data Format 5 (HDF5) file format and Gzip compression, providing techniques for file size reduction [12]. Mushtaq et al. evaluate symmetric encryption algorithms, highlighting Advanced Encryption Standard (AES), Blowfish, and HiSea as preferred choices [15]. Additionally, Jaspin, Selvan, and Sahana propose a double encryption mechanism using AES and Rivest–Shamir–Adleman (RSA) algorithms, emphasizing multi-layered security approaches [16]. Despite these advancements, further research is needed to comprehensively evaluate EM data compression using existing compression techniques and the behavior of uploading time and the computer resources when using different chunk sizes for the chunk uploading mechanism.

III. Design and Implementation

This work entails developing a software platform, called *EMvidence*, for EM-SCA, aimed at aiding digital forensic investigators in real-life scenarios. The platform consists of two main components: (1) Data acquiring application and (2) Web application. The former is a desktop tool for acquiring EM data with minimal configuration, while the latter utilizes cloud services for processing data and generating insights. This methodology ensures usability and efficiency in EM data acquisition, processing, and analysis, catering to the needs of forensic investigations.

A. EM Data Acquisition

EM data acquisition in the context of EM-SCA involves capturing and recording EM radiation from computing devices during their operations. This process is pivotal as the captured data serves as the foundation for analysis and insight generation in for investigations. An SDR device-based approach was deemed optimal for EM acquisition due to its ability to capture a broad range of frequencies and its cost-effectiveness. The HackRF One SDR hardware device, with its capability to capture EM emissions in the frequency range of 1 MHz to 6 GHz and a maximum sampling rate of 20 MHz, was selected for this purpose [17]. Additionally, higher sampling rates enhance the quality of the captured EM data, though this also results in larger file sizes.

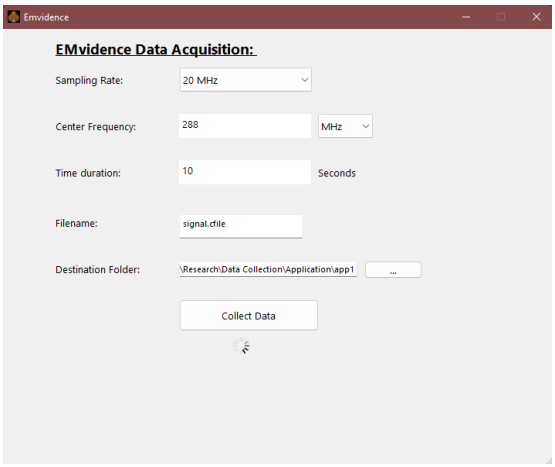

Fig. 1: User Interface (UI) of EMvidence data acquiring application

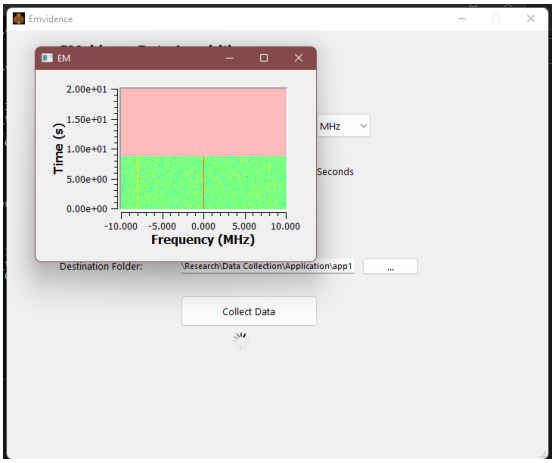

Fig. 2: UI of EMvidence data acquiring application when collecting data

To interact with the HackRF One hardware device, the GNU Radio Python library was chosen for its capabilities

in capturing and recording EM data, along with Python as the programming language due to its compatibility with the GNU Radio library and suitability for desktop application development. The hardware setup comprises the HackRF One device, a near-field H-loop antenna for capturing EM waves, a host computer for configuration and data storage, and an IoT device under investigation. An Arduino UNO device was taken as a representative IoT device in evaluations. Positioning the antenna close to the microcontroller chip of the target device enhances signal strength of the data being acquired for better data analysis.

The data acquiring application is designed to operate seamlessly on both Windows and Linux operating systems, ensuring portability and versatility for digital forensic investigators. Its UI, depicted in Figure 1, allows investigators to set input parameters and initiate data acquisition. During the acquisition process, a spectrogram visualises the EM data being captured (see Figure 2).

B. File Handling on the Client Side

Handling of EM data files on the client side in the EMvidence web application is a critical aspect that ensures the secure and efficient uploading of large EM data files to the server. When implementing the file upload feature, several challenges related to data integrity and confidentiality must be addressed to ensure the security of EM data. Upon initiating the upload process, the user is presented with a file upload interface, where they enter the required information, such as device name, center frequency, sampling rate, and the EM data file, before clicking the upload button to begin the process. Once the user triggers the upload process, the file undergoes several processing steps to ensure integrity, confidentiality, and efficiency. First, the MD5 hash value of the file is calculated to verify data integrity during transmission and storage. Next, the file is compressed using the Gzip compression technique to reduce its size, optimizing bandwidth utilization and minimizing upload times.

After the initial preprocessing, the chunk upload mechanism is employed to transmit the EM file to the server via an end-to-end encrypted channel as shown in Figure 3. This mechanism involves dividing the file into smaller chunks or segments for easier management and parallel uploading. If any segment fails to upload or becomes corrupted during transmission, the client re-transmits that specific chunk, minimizing data loss and improving reliability. Once all chunks are successfully uploaded and assembled on the server side, the user receives confirmation of the successful upload.

C. File Handling on the Server

Upon receiving uploaded file chunks from temporary storage, the server initiates the process of reconstructing the original file by arranging the chunks in the correct order and combining them to recreate the file in its entirety. Once the file upload is complete, a scheduled task is triggered on the server side to convert the uploaded file into a suitable format for analysis. This task involves a series of processing steps aimed

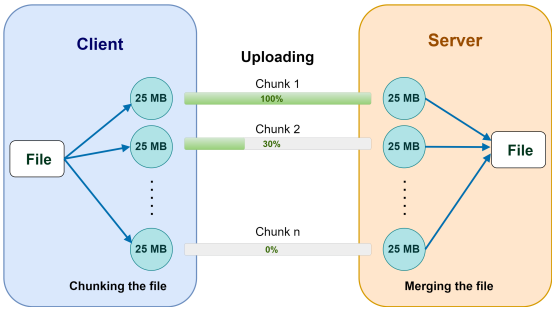

Fig. 3: Chunk upload mechanism

at preparing the file for subsequent analysis stages. Firstly, the server decompress the received file to its original state. Then, it recalculates the MD5 hash of the decompressed file for integrity verification. Finally, the file is converted into the HDF5 format.

The HDF5 format enables the compressed file content to remain accessible without decompression, facilitating efficient storage and retrieval. Subsequently, the compressed HDF5 file undergoes encryption using the AES algorithm to safeguard its content during storage until it is used to generate forensic insights after processing with analysis plugins.

D. EM Data Pre-processing

The stages of pre-processing encompasses both mandatory and optional steps, some of which can be customized by investigators during their investigative process. The key stages include down-sampling, Fourier transformation, and sample selection. Down-sampling is employed to decrease file size, thereby easing CPU and memory usage during EM file processing. Investigators can choose various down-sampling rates or opt not to downsample the data during analysis. Afterwards, the EM data in HDF5 format is converted into NumPy arrays to simplify pre-processing and subsequent analysis steps.

Fourier transformation is then applied to convert time-domain data into the frequency domain, ensuring consistency across multiple EM traces for effective ML analysis. Sample selection further enhances result accuracy and reduces memory overhead during processing by selecting specific time components for analysis. Users can choose between predefined options or proceed without sample selection. The pre-processed data undergoes format transformation to convert it into a suitable format for processing. The pre-processed NumPy array is then saved as a .npy format file, serving as input for EM data analysis using ML based EM-SCA plugins, ultimately leading to the generation of forensic insights.

E. Plug-ins for Analysis

The platform is structured to provide a standardized workflow from the development, packaging, and integration of plugins designed to extract forensic insights from EM data (see Figure 4). The design phase starts with identifying forensic requirements and IoT devices to generating forensic insights and addressing unmet needs. Following the design phase, the

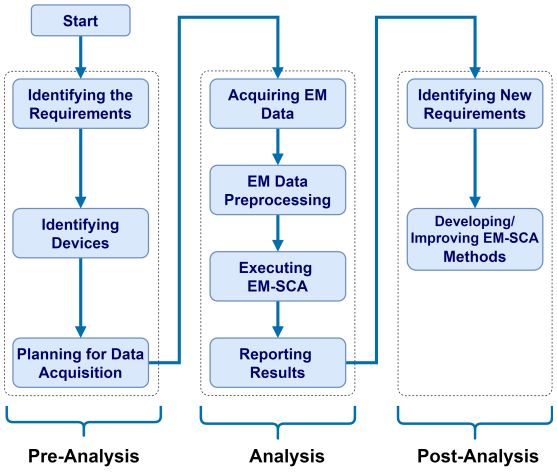

Fig. 4: A probable investigative workflow for IoT forensics using EM-SCA methods.

plugin creation procedure involves making initial decisions on the type of IoT device and desired forensic insight, followed by dataset generation to input into ML models and ML model training. The plugin package consists of components such as the ML model, Python script, dependency file, and plugin icon.

Developers can upload their plugins through an interface in the web application, providing details such as plugin name, associated center frequency, device information, and files including the ML model and Python script. Admins then review and approve uploaded plugins, ensuring only verified and reliable plugins are integrated into the platform. The platform opts for a centralized approach to streamline deployment and ensure economic viability. Additionally, the integration of a rating mechanism from users and the forensic community further enhances the reliability and accuracy of integrated plugins, fostering continuous improvement and innovation within the EMvidence platform.

IV. Evaluation and Results

A. Evaluation of Data Acquisition

The evaluation involved acquiring data from the EMVidence data acquisition application for variable time periods (10s, 20s, 30s) at two sampling rates (10 MHz, 20 MHz), using the theoretical file size for each time period and sampling rate as a reference. Higher sampling rates and longer data collection periods resulted in larger file sizes, as more detailed information and data were captured over time. However, a difference was observed between the theoretical and experimental file sizes, suggesting potential influences from internal and external factors during the data collection process.

B. Evaluation of File Transfer

The evaluation encompassed three critical aspects. Firstly, the comparison of Gzip and bzip2 compression algorithms for client-side file processing was performed (see Figure 5). Despite bzip2's commendable compression ratio, Gzip offered a notable balance between compression ratio, compression time, and resource usage, making it the preferred choice for compressing EM data. Subsequently, the evaluation scrutinized the chunk-uploading mechanism, focusing on the impact of varying chunk sizes on upload time and resource consumption. Surprisingly, no significant differences were detected across different chunk sizes, suggesting that the choice of chunk size may not significantly affect upload efficiency in this context.

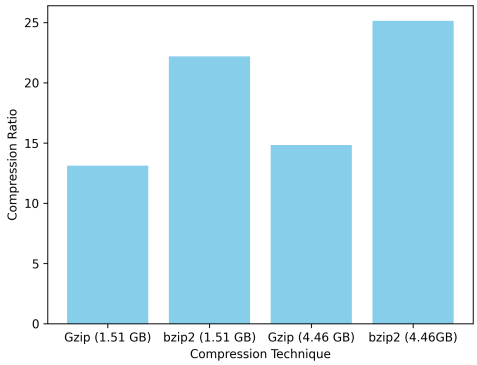

Fig. 5: Compression ratio of EM data files according to different types of compression algorithms

Despite the chunk upload mechanism's longer upload times (see Figure 6) due to server-side chunk merging processes, its resource efficiency (see Figures 7 and 8) makes it a preferable option, particularly in scenarios of simultaneous user uploads. Additionally, the chunk-uploading mechanism's reliability in mitigating network-related failures underscores its suitability for handling large EM datasets. Combining the chunk uploading mechanism with Gzip compression emerged as the optimal strategy, aligning with other implementation choices aimed at effectively managing EM datasets and facilitating forensic insights through EM-SCA.

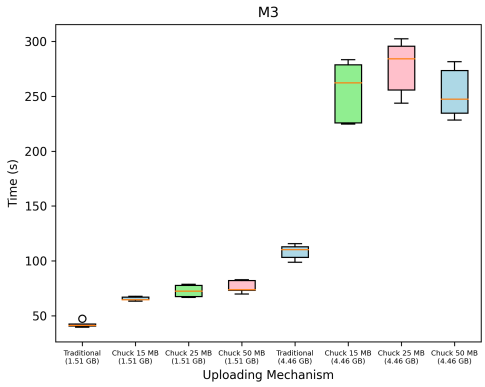

Fig. 6: Time which it takes to upload different file sizes using traditional and chunk uploading methods

C. Evaluation of Data Preprocessing

The evaluation of EMvidence's data pre-processing configurations revealed crucial insights into their impact on CPU

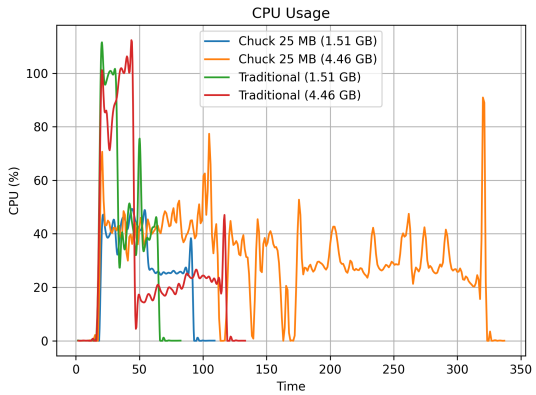

Fig. 7: CPU usage for different file sizes using traditional and chunk uploading methods

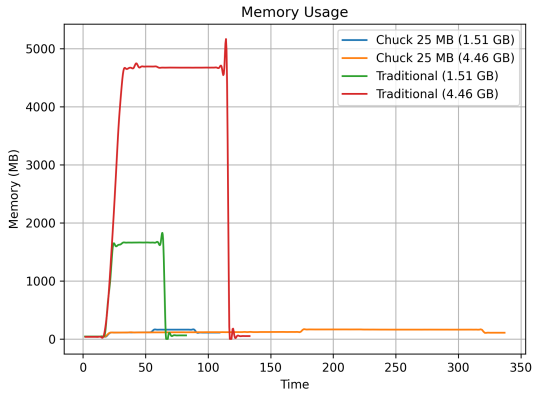

Fig. 8: Memory usage for different file sizes using traditional and chunk uploading methods

utilization, memory usage, file size variation, and processing time. The evaluation was conducted for 8 combinations of options for pre-processing steps (named S1–S8) across 3 sandbox configurations (namely M1–M3) with varying CPU and memory availability. The sandbox configuration with 2 CPU cores and 2 GB memory gave timeout errors for all the combinations that contain down-sampling pre-processing steps, while the combinations that do not include down-sampling completed the pre-processing steps. This indicates that down-sampling is the most CPU and memory-intensive task out of all pre-processing steps.

Figure 10 illustrates the substantial impact of pre-processing on reducing file sizes, with configurations S6 and S8 showing the most significant reductions. This is where the sampling selection process is applied in addition to down-sampling. However, it's worth noting that configurations not involving down-sampling & sample selection, such as S1 and S3, had the lowest impact on file size reduction. Additionally, processing time varied considerably across configurations, with down-sampling configurations (S5 to S8) exhibiting longer processing times compared to non-down-sampling configurations. (Refer Figure 9) This underscores the trade-off between processing time and file size reduction, emphasizing the need for users to carefully select pre-processing configurations based on their specific requirements and available resources.

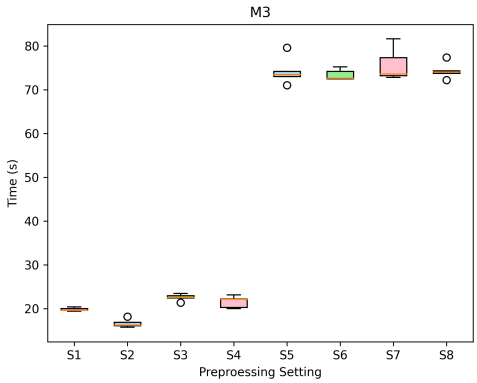

Fig. 9: Processing time of pre-processing steps in M3 sandbox (8 CPU cores and 8GB memory)

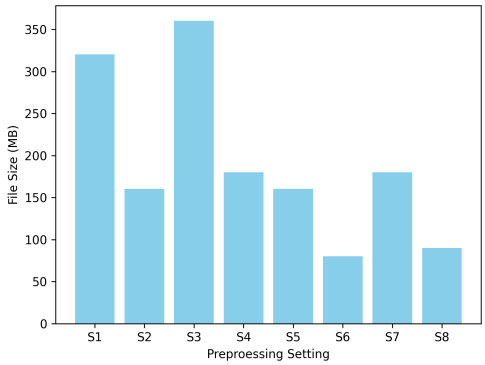

Fig. 10: File size variation of the pre-processing steps

D. Monolithic vs. Distributed Plugin Architectures

While both centralized and distributed plugin management architectures demonstrated similar CPU utilization, the distributed approach showed advantages in memory usage and processing time. However, qualitative factors such as data confidentiality and deployment costs must also be considered when determining the most suitable plugin management architecture. Due to these qualitative factors, it is eminent to go with the centralized EM-SCA plug-in architecture at the moment. These evaluation results provide valuable insights for optimizing EM-SCA data analysis processes within the EMvidence platform, contributing to enhanced efficiency and performance in forensic investigations involving EM data.

V. Conclusion

This research focused on enhancing the handling of large EM datasets for forensic analysis through EM-SCA. By introducing a FaaS platform, the project addressed challenges in

managing EM data on both the client and server sides. Through secure file handling processes and comprehensive evaluations, key methodologies were established. The seamless integration of data acquisition and pre-processing into the EMvidence platform has significantly improved the efficiency of the EM-SCA process in digital forensics. Future improvements to the framework include extending compatibility of the EMvidence data acquiring application to support SDR devices beyond HackRF One, implementing and evaluating EM-SCA plug-ins for commonly encountered IoT devices, and developing a mechanism for efficient file chunk concatenation. These enhancements will further refine the platform's versatility and effectiveness.

REFERENCES

[1] S. Soltani and S. A. H. Seno, "A survey on digital evidence collection and analysis," in *2017 7th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2017, pp. 247–253. DOI: 10.1109/ICCKE.2017.8167885.

[2] G. Horsman, "Acpo principles for digital evidence: Time for an update?" *Forensic Science International: Reports*, vol. 2, p. 100 076, 2020.

[3] F. Courbon, S. Skorobogatov, and C. Woods, "Reverse engineering flash eeprom memories using scanning electron microscopy," in *Smart Card Research and Advanced Applications*, K. Lemke-Rust and M. Tunstall, Eds., Cham: Springer International Publishing, 2017, pp. 57–72, ISBN: 978-3-319-54669-8.

[4] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the internet of things (iot) forensics: Challenges, approaches, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1191–1221, 2020.

[5] D. Lillis, B. Becker, T. O'Sullivan, and M. Scanlon, "Current challenges and future research areas for digital forensic investigation," May 2016. DOI: 10.13140/RG.2.2.34898.76489.

[6] E. Peeters, F.-X. Standaert, and J.-J. Quisquater, "Power and electromagnetic analysis: Improved model, consequences and comparisons," *Integration*, vol. 40, no. 1, pp. 52–60, 2007.

[7] A. Sayakkara, N.-A. Le-Khac, and M. Scanlon, "Leveraging electromagnetic side-channel analysis for the investigation of iot devices," *Digital Investigation*, vol. 29, S94–S103, 2019.

[8] *About gnu radio*. [Online]. Available: https://www.gnuradio.org/about/.

[9] *Software-defined radio*. [Online]. Available: https://en.wikipedia.org/wiki/Software-defined_radio.

[10] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *Cryptographic Hardware and Embedded Systems—CHES 2001: Third International Workshop Paris, France, May 14–16, 2001 Proceedings 3*, Springer, 2001, pp. 251–261.

[11] L. N. Nguyen, C.-L. Cheng, F. T. Werner, M. Prvulovic, and A. Zajic, "A comparison of backscattering, em, and power side-channels and their performance in detecting software and hardware intrusions," *Journal of Hardware and Systems Security*, vol. 4, pp. 150–165, 2020.

[12] A. P. Sayakkara and N.-A. Le-Khac, "Electromagnetic side-channel analysis for iot forensics: Challenges, framework, and datasets," *IEEE Access*, vol. 9, pp. 113 585–113 598, 2021.

[13] C. Miller, D. Glendowne, D. Dampier, and K. Blaylock, "Forensicloud: An architecture for digital forensic analysis in the cloud," 2014.

[14] H. Mallafi, "Performance analysis in web based data uploading using lz77 compression and chunking method," *Indonesia Journal on Computing (Indo-JC)*, vol. 1, no. 1, pp. 37–48, 2016.

[15] M. F. Mushtaq, S. Jamel, A. H. Disina, Z. A. Pindar, N. S. A. Shakir, and M. M. Deris, "A survey on the cryptographic encryption algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 11, 2017.

[16] K. Jaspin, S. Selvan, S. Sahana, and G. Thanmai, "Efficient and secure file transfer in cloud through double encryption using aes and rsa algorithm," in *2021 international conference on emerging smart computing and informatics (ESCI)*, IEEE, 2021, pp. 791–796.

[17] *HackRF One - Great Scott Gadgets*. [Online]. Available: https://greatscottgadgets.com/hackrf/one/ (visited on 09/06/2023).