CSAW ESC 2024 Final Report

GT Security Innovators

Divyen Marsonia
Georgia Institute of Technology
Atlanta, Georgia
dmarsonia3@gatech.edu

Jennifer Maaskant
University of North Georgia
Atlanta, Georgia
jrmaas3420@ung.edu

Darshan Singh
Georgia Institute of Technology
Atlanta, Georgia
dsingh93@gatech.edu

Abstract— The 2024 Embedded Security Challenge (ESC) is an international competition focusing on hacking into the hardware of embedded systems. This year's ESC is on using side-channel attacks (SCA) on cyber-physical systems (CPS) in the manufacturing industry. As Cyber-Physical Systems (CPS) become increasingly complex, they become more susceptible to side-channel attacks (SCAs), which exploit unintended information leaks to breach security. This pager explores the various side-channel attack methodologies applicable to CPS, focusing on acoustic, power analysis, timing, electromagnetic emissions, and encryption attacks. It provides an overview of these attacks, detailing how they exploit specific vulnerabilities in various CPS components that run off Arduino-based systems, such as sensors, actuators, and controllers. In response to these threats, this pager explores essential mitigation strategies to protect these systems from SCAs, including obfuscating power consumption, obscuring timing information, and shielding electromagnetic emissions. Highlighting attack methods and mitigations, this research aims to understand the risks posed by side-channel attacks in CPS and offer recommendations for improving system security.

I. INTRODUCTION

Cyber-Physical Systems (CPS) integrate technology and physical systems to interact with the real world. CPS includes sensors, chips, computing, control systems, motors, wireless connectivity, and more. All these parts act synonymously to compute in real-time to make decisions, improving automation and efficiency. CPS uses sensors to collect data from the environment, which unlocks the ability to respond to environmental changes, which was previously impossible. One example of CPS is self-driving cars. They use sensors such as cameras, lidars, and radar to collect information about the environment and process it in real time to control the vehicle. CPS is also used in medical devices, such as a pacemaker to monitor a patient's heart and adjust the rhythm. Other examples include nuclear power plant systems, control, railway tracks, traffic lights, and airplanes.

Securing cyber-physical systems is important because they control critical infrastructures like manufacturing, power plants, water facilities, transportation, and airports. There are 16 critical infrastructures described by the Department of Homeland Security viral to national security, many of which are controlled by CPS. For example, in the energy sector, CPSs are used in smart grids for power distribution, renewable energy sources integration, and electricity demand surges. Additionally, the exportation of these systems can impact human lives. As CSP becomes more intertwined with our real world, vulnerabilities could lead to chaos, in the physical world, not just cyber. For example, a successful attack on a CPS controlling a power grid could lead to widespread blackouts, affecting millions and causing economic damage. Another example is if bad actors exploit pacemakers, they could harm the patient. This affects not one endpoint but the whole network.

One specific attack vector on CPS is the side-channel attack. This attack takes advantage of unintended information leakage in the physical electrical circuit. This is very different from exporting software or network protocols. Side-channel attacks use power consumption, electromagnetic emissions, or timing information to find sensitive data. Due to the nature of the systems, the surface area is wide and easily accessible.

II. ECS HARDWARE CONFIGURATION

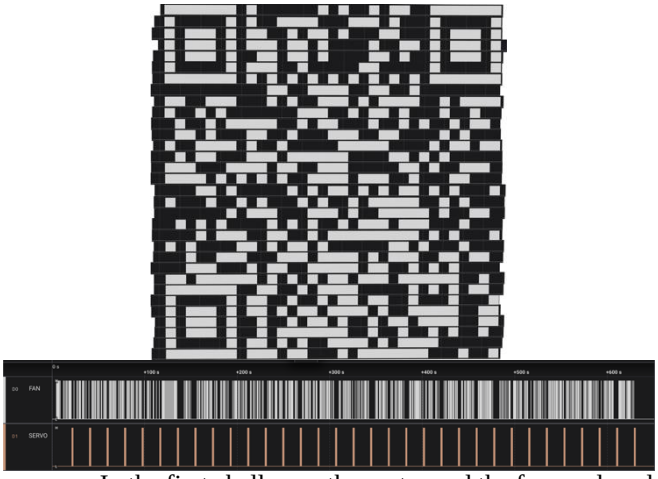
This year’s ESC included the following components: ELEGOO UNO R3 Board ATmega328P with USB Cable, 12 Voltage 2A 24W Switching Power Supply, L293D DC Motor Drive Shield Expansion Board, Stepper motor 12V 350mA, MG996R Servo Motor 360 control angle and 120mm DC 12V Computer Fan.

At the heart of this challenge is the Arduino-based computer with an ATmega328P chip executing the instructions programmed into it. The device is powered by a 12-volt power supply, with its wires stripped and connected directly to the motor shield. This is to prevent burning the USB ports on the external computer when programing the board with the Arduino IDE. L293D DC Motor Drive Shield Expansion Board is attached on top of the board to connect the Stepper Motor, Servo Motor, and computer fan. UNO R3 Board is Equipped with 14 digital pins to control these components for inputs and outputs. This year's challenge has four weeks with each week having two parts with instructions and the hex file to run the challenge. Further, we will discuss how these

connected components will be exploited using specific side-channel attacks.

III. Week One Challenges

Normal Or Though



In the first challenge, the motor and the fan produced noises for 10 minutes. A logic analyzer was used to capture the transmitted signal. Signals to the servo motor and the fan were collected using Logic2. This revealed 33 timing bars from the servo motor's revolutions, suggesting a 33 x 33 QR code format. To decode the pattern, 33 screenshots of dashes from the fan were taken and stacked on top of each other in Canva. The image was expanded horizontally and vertically to reconstruct the QR code. After scanning the QR code, it revealed RQ matrix. Using the hints from post competition, the QR code should have led us to a Google form with "Kw1CkRe5p0Nze" as the flag.

Friendly Disposition

All phases of the Friendly Disposition challenge employed the same process for solving each step. During each phase the motors rotated and provided a set of ASCII values, then the motor rotated without displaying the corresponding ASCII values. By testing all possible inputs within each phase and measuring the output using a logic analyzer, it was possible to construct a key to assign motor signals to ASCII values in order of smallest to largest. This could then be applied to construct a sequence of numbers that could be matched to a preexisting sequence of numbers, to a point. By mapping out the values from the device to the preexisting sequence, and noting how they diverged, it was possible to determine the modulus factor the values from the motors were offset by, which could then be used to convert the next values in the preexisting sequence to values that could be used for input.

In this way, phases 1 through 4 were solved. Phase 1 used the Fibonacci sequence with a modulus of 26 and had a flag of "K J U E Z E E J". Phase 2 used the prime powers sequence with a modulus of 10 and had a flag of "1, 2, 7, 1, 3, 7, 9, 3, 9, 1, 4, 7, 1, 3, 9, 1". Phase 3 used the Mersenne prime numbers sequence with a modulus of 26 and had a flag of "s, s, o, c, q, i, g, u". Phase 4 used Narayana's cows sequence with a modulus of 16 and had a flag of "% & # (. !) ' ( ) " ".

The final step used all possible inputs from the prior sequences, and did not print motor values as the other phases had.

I. Week Two Challenges

KeyRing 1

```python
def calculate_similarity(wav1, wav2):
    # Ensure both audio files have the same length by padding or trimming
    min_length = min(len(wav1), len(wav2))
    wav1 = wav1[:min_length]
    wav2 = wav2[:min_length]
    mse = tf.keras.losses.MeanSquaredError()(wav1, wav2)
    return mse.numpy()
```

This part of the challenge required classifying what type of key is being printed using the audio from the 3-D printer. Using the provided audio files classified as KeyA, KeyB, KeyC, KeyD. The approach was to create a signature to match a classified audio file to an unclassified audio file. Google Collab was used to streamline the audio classification process with TensorFlow and related libraries. Necessary packages included: TensorFlow TensorFlow-io, and matplotlib. A mounted cloud drive was used to access classified and unclassified audio files. A function was implemented to measure the Mean Squared Error (MSE) metric for similarity. The function is shown above in the image. The resulting similarity scores indicated the match level between each labeled key and the unknown audio sample. Using this data be compared, which audio file was the closest match to a specific key and classified it as following in the table.

A	B	C	D
Sample	Key	MSE	Best Match
sample3.wav	KeyC.wav	0.027831037	KeyC.wav
sample4.wav	KeyC.wav	0.019619463	KeyD.wav
sample5.wav	KeyC.wav	0.02059058	KeyC.wav
sample6.wav	KeyC.wav	0.017496413	KeyC.wav
sample7.wav	KeyC.wav	0.028446302	KeyC.wav
sample8.wav	KeyC.wav	0.017263187	KeyD.wav
sample10.wav	KeyC.wav	0.029220194	KeyC.wav
sample13.wav	KeyC.wav	0.019301603	KeyD.wav
sample14.wav	KeyC.wav	0.013824704	KeyC.wav
sample17.wav	KeyC.wav	0.027974512	KeyD.wav
sample19.wav	KeyC.wav	0.015576437	KeyD.wav
sample20.wav	KeyC.wav	0.024820909	KeyC.wav
sample24.wav	KeyC.wav	0.027656555	KeyD.wav
sample25.wav	KeyC.wav	0.014398677	KeyD.wav
sample26.wav	KeyC.wav	0.020109305	KeyC.wav
sample29.wav	KeyC.wav	0.024199847	KeyC.wav
sample35.wav	KeyC.wav	0.022275995	KeyC.wav
sample36.wav	KeyC.wav	0.020146731	KeyD.wav
sample37.wav	KeyC.wav	0.021179294	KeyC.wav
sample40.wav	KeyC.wav	0.018124135	KeyC.wav

IV. Week Three Challenges

Lizzy

The Lizzy challenge presented a hex file, Lizzy.hex, as the starting point. Objcopy was used to convert the hex file to binary format, after which objdump was applied to disassemble the file for further examination. In the disassembled code, a range of readable strings were identified, which provided clues into the challenge's nature.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ strings -n 13 Lizzy.bin
/*********************************************************************/
Turning Counterclockwise 360 degrees ...
Turning Clockwise 360 degrees ...
Enter Counterclockwise Calibration Integer
Enter Clockwise Calibration Integer
/********************  YOU BEAT THE CHALLENGE!!!   ********************/
/********************          Congrats!!!         ********************/
Bang clang Maxwell
s silver hammer made sure that she was dead. Try Again.
Correct Flag!! Put the flag and sequence encodings in your report!!
@D%U%f%w%T.G/v/e/U-
P#/4/E/V/g/x/
Obfuscation is invalid. Please contact challenge administrator.
Welcome to Lizzy!
/2 (default 100):
Calibrated to
LaLaLa! Type in your response. Make sure to use same protocol/cipher:
```

Among these strings, phrases like "Turning Counterclockwise 360 degrees" and "Calibration" indicated that the hex file likely manages motor control, with calibration suggesting that the motor's precise movements must be set as part of the operation. Additionally, phrases such as "Type in your response. Make sure to use the same protocol/cipher" hinted at the need for an encryption protocol, while "Obfuscation is invalid" implied that the challenge might involve recognizing or handling code obfuscation techniques.

Given the hex file's control of the motor and the reference to encryption protocols, it is likely that the challenge simulates an acoustic side-channel attack. In such an attack, the motor's operational sounds could leak information about its activity or encoded commands. The task would then involve analyzing these sounds, interpreting the obfuscated or encrypted signals, and successfully calibrating the system to respond correctly, completing the challenge.

FAST & MAX

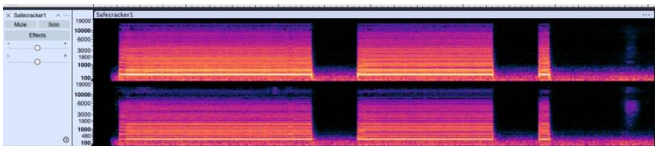
The Fast & Max challenge presented a unique scenario, combining cryptographic techniques and motor control in a secure bank vault system simulation. The challenge provided two hex files: FastMax.hex, the main challenge file, and FastMax_Dummy.hex, specifically included for reverse engineering to learn about the control logic.

Using Ghidra to analyze FastMax_Dummy.hex revealed extensive code obfuscation, though certain helpful strings were found. Phrases like "What is the employee card number?", "Please enter the Bank PIN:", and "Employee Card Number Accepted!" implied a two-stage access system. In this system, an employee ID and a bank PIN were required to unlock two safes, with the ID encoded using RSA-like encryption and the PIN limited to alphabetic characters.

The identified strings also pointed to hardware interaction. Statements such as "Shaft in position!!" and "Incorrect Rotation Amount" suggested that the hex file controlled a motor. These motor operations served as an acoustic attack side channel, allowing analysis of motor sounds in response to entered codes to deduce patterns associated with the correct ID and PIN.

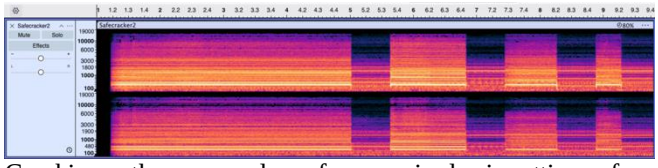
V. WEEK FOUR CHALLENGES

Safecracker 1



This challenge required putting three combinations of a code to crack the safe. The code combination is a number followed by the rotation type. For Example, 100S, 120S. S meaning single. Other rotation types include double(D), Interleave (I), Micro step (M). It was given that the code will be in S mode. An audio file Safecracker1.mp3 was given, which captured the sound of the lack rotations. Importing the file in Audacity, switching the view to spectrogram, and measuring the sound's length indicated that three dashes with timings 4.2 seconds, 3 seconds and 250ms were captured. By analyzing the demo audio file and the safecracker_release.ino it was determined that 3.2 seconds is 100S. This is based on code "run_motor(100, FORWARD, SINGLE);" and the demo audio, which had a length of the motor spinning for 3.2 seconds. The approach was to re-create these timings by inputting corresponding combinations. 100/3.2 is the base unit equaling 31.25. So, the (4.2s,3s and 250ms) correspond to 131.25, 93.75 and 7.8125. These combinations did not crack the safe so the second approach was to try all combinations rounding up and down, which also did not work. The following combination was attempted. 131, 93, 7 131, 93, 8 131, 94, 7 131, 94, 8 131, 95, 7 131, 95, 8 132, 93, 7 132, 93, 8 132, 94, 7 132, 94, 8 132, 95, 7 132, 95, 8 133, 93, 7 133, 93, 8 133, 94, 7 133, 94, 8 133, 95, 7 133, 95, 8 134, 93, 7

SAFECRACKER 2



Cracking the second safe required inputting four combinations. This challenge introduced four distinct modes: Single (S), Double (D), Interleave (I), and Micro step (M), along with corresponding numbers, such as 100S, 100D, 100I, and 100M. Analyzing the second audio file indicated four timing bars with length 7.6s 2.4s 1.6s 0.8s corresponding to numbers 237.5, 75, 50, 25. The second part of the challenge was to determine the mode that goes with these numbers. Using the demo audio file, which had all three modes and had very distinct sounds. Single was smoothing, Double was high pitch, interleave was medium pitch and micro step was filled with base. Then again, playing Safecracker2.mp3. The first motor spin was classified as interleave, then double followed by two single spins. After classifying these the combination was determined to be 237I, 75D,50S,25S.

VI. SIDE CHANNEL ATTACKS IN CPS

Side-Channel Attacks (SCAs) exploit unintended information leakage from a system during its normal operations. These attacks pose a large risk to CPS due to the close integration between digital controls and physical processes. CPS rely on interconnected networks of sensors, controllers, and actuators to monitor and manipulate physical machinery, effectively bridging the gap between the cyber and physical worlds. While this integration enhances efficiency, it also creates vulnerabilities that SCAs can exploit by targeting physical components.

SCAs are particularly dangerous because they can manipulate or extract data from a system without directly interacting with its software, often bypassing conventional security measures. A successful SCA can have serious repercussions in manufacturing environments, including unauthorized access

to sensitive data, disruption of production lines, and even safety risks for operators and equipment.

VII. Attack Methodologies and Mitigation Strategies

Acoustic attacks

Acoustic attacks exploit noise leakage from devices as they are used to perform operations, which can be used to reproduce the secrets being generated on the device. These attacks date back to the 1950s, such as Operation ENGULF, in which British intelligence operations used audio recordings of Egyptian cipher machines to recreate the encryption settings used [12].

One modern example of this is using the sound from a keyboard or keypad to reproduce the keystrokes and extract secrets. By noting the target device and keyboard and gathering data on the sounds it emits during use, it is possible to determine what is being typed. With advancements in microphone technology, and with enough data, it has been proven possible to reproduce keystrokes from a device as simple as a smartwatch [14].

Another example of this is using the sounds from an additive manufacturing system, such as a 14D printer, to recreate the objects being produced [15]. This attack would mainly be used to spy on manufacturers or to steal intellectual property with considerably less effort, as by using microphones to capture audio the attack could be done remotely from anywhere with a suitable connection.

Acoustic attacks Mitigation

One mitigation strategy could be to dampen the noise and vibrations coming from the device as best as possible. This could be accomplished with acoustic foam or similar material, and by taking special care to maintain the devices so they experience minimal wear and tear. While this would lessen the attack surface of the machine quite a bit, it would be rather costly to implement, as such a mitigation would have to cover the entire sound spectrum to be truly effective. In a large manufacturing context, this could very quickly grow expensive to implement and maintain [15]. Another mitigation could be to mask the noise with that of a louder noise at a similar frequency, or to mix in the sounds of random inputs at random intervals. In this way, it would be more difficult to pick out useful. While playing white noise while typing was demonstrated to mitigate the effectiveness of the attack on key inputs, it was found that mixing in random inputs was generally more effective [13]. A final mitigation for manufacturing systems could be to spread out loads on each machine as equally as possible, so that it is more difficult to determine what is being produced on any single machine. The difficulty with this is that it would require restructuring machines to accommodate the new load, which could be costly on a large scale [15].

Power Attacks

Power attacks leverage the variations in power consumption during a system's operation. By measuring the power usage patterns of a CPS device, attackers can discover sensitive information such as cryptographic keys or algorithms. These types of attacks can be divided into three types:

1. Simple Power Analysis (SPA) involves directly observing the overall power consumption patterns of the system. By detecting large-scale variations in power usage, attackers can reveal broad information about the operations being performed, such as which algorithms or functions are in use.

2. Differential Power Analysis (DPA) is a more complex method that uses statistical techniques to analyze subtle differences in power consumption throughout varying operations. These variations can provide information by correlating power consumption with known inputs or outputs.

3. Correlation Power Analysis (CPA) refines DPA by focusing on the correlation between power consumption and specific values within the system, such as bits in a cryptographic algorithm. By statistically modeling and comparing the power consumption traces with hypothetical values, attackers can have higher accuracy while making predictions about data.

Power Attacks Mitigation

Possible mitigations against power attacks depend on the type of attack being used against the system. Possible mitigations against SPA, DPA, and CPA attacks would be to obfuscate the system's power use, or to combine all of the steps in each round of encryption into a single iterative function [5].

Timing Attacks

Timing attacks exploit differences in the time a system takes to execute specific operations. These attacks often focus on cryptographic systems, where slight variations in execution time can unintentionally reveal key information. One example is RSA cryptanalysis, where an attacker monitors the timing of decryption operations. The time required for these operations can differ based on the private key bits. By repeatedly testing different passwords and measuring how long each operation takes, an attacker can gather statistical data to infer the private key. For instance, if a private key bit is 1, the operation may take longer than if the bit is 0. Repeating this process enables the reconstruction of the entire private key.

One real word example happened in 2003 on OpenSSL implementation of RSA encryption. This was discovered by two reacher servers from Stanford, where they exploited the vulnerability in the Chinese Remainder Theorem (CRT) and Montgomery multiplication for RSA decryption, which led to the leaked timing information[16]. First step involved sending crafted text messages to the and measuring the decryption time. After about doing 1000,000 queries and statistical analysis they were able to extract the 1024-bit RSA private key from an OpenSSL 0.9.7 server

Timing Attacks Mitigation

The mitigation strategy against the OpenSSL attack is to implement RSA blinding, which introduces randomness into the decryption timing, effectively making timing attacks useless. This works by first multiplying the input text by a random value (r) raised to the public exponent (e): C' = C * r^e mod n. Then to perform the actual description, the result

is by r mod n to obtain the original plaintext. This operation creates a timing difference, hiding the timing information. [17]

Another possible mitigation against timing attacks is to standardize the amount of time each operation takes to execute, so that a secret operation does not impact on how long a resource is being used [3]. If implemented properly, this would prevent various side channel attacks from the CSAW ESC 2023 competition such as the Spitfire challenge, which used a timing attack to analyze the clicks of the relay to determine the rising and falling edge of the signals [6].

Electromagnetic Attacks

Electromagnetic attacks measure and analyze the electromagnetic emissions produced by electronic components within the system during operation. With specialized equipment, attackers can capture these emissions and reverse-engineer the data being processed to compromise the integrity of the entire system. One example of this is found in the Intel Atom series of CPUs, which were found to leak RSA and AES secret keys on generation [3].

Especially in cryptographic systems where a induction core can be used to capture the electromagnetic waves to decipher the operations and data. This exactly happened in the Intel Atom processors where a frequencies between 50 MHz and 85 MHz revealed the RSA and AES encryption operations similar to the timing attack. The signal is captured and converted into digital and then with some advanced signal processing techniques the information is extracted. A real-world example was showcased in 2014 by Tel Aviv University where they extracted the 4096-bit RSA encryption keys from a laptop capturing the electromagnetic signal from the CPU. First, they exposed the computer chassis and then probed it with an electromagnetic capture device and used ethernet cables to transfer the digital signal. Once the signal was captured and sent over, it was just matter of analyzing and cracking the key. Most surprising part of this demonstration was the distance where up to 50 cm from the target device and it was still able to execute the attack. [18]

Electromagnetic Attacks Mitigation

A possible mitigation against Electromagnetic Attacks is to use some form of electromagnetic shielding to protect against electromagnetic emissions leaking out of the system. This could be accomplished using a Faraday cage, preventing electromagnetic signals from entering or exiting the system [7].

Encryption Attacks

Encryption attacks exploit the vulnerabilities in cryptographic algorithms to gain access to data within CPS. One such vulnerability involves hash collisions, where two distinct inputs generate the same hash value, enabling attackers to bypass security controls. A notable example is the CSAW ESC 2023 "All White Party" challenge, which used the insecure SHA-1 hashing algorithm. In this case, only the first five bytes of the hash were checked, making it susceptible to a hash collision attack. Another example of this is by using rainbow tables, which can be used to decipher hashed passwords more easily given a list of hashed passwords [4].

Encryption Attacks Mitigation

A possible mitigation against encryption attacks is to use more robust encryption methods and not rely on encryption methods that have disclosed vulnerabilities. In practice, this would mean using SHA-256 or SHA-3 over SHA-1 and MD5, as both SHA-1 and MD5 have been demonstrated to be vulnerable to file collisions. While SHA-1 is more difficult to generate a collision for, as the first proven instance of it being broken took the equivalent processing power as 6,500 years of single-CPU computations and 110 years of single-GPU computations [8], MD5 has been known to be vulnerable to hash collisions since 2004 [9], and even has tools written for generating MD5 hash collisions [10]. The capability to generate SHA-1 hash collisions will only grow as computing power grows more accessible.

VIII. Individual contributions

Divyen Marsonia contributed to the introduction of CSP's and the abstract, what they are and why they are important. Additionally, they created a detailed report on how this year's competition Arduino board will be configured and outlined each component. Also ensured the creation of documents and presentation slides in a well-formatted manner. Near the end of the semester, they edited the conclusion and expanded attack and mitigation strategies for timing, power and encryption attacks. Then we restructured the paper to follow a good flow of ideas. During the competition phase, they focused on visiting the lab to work on the hardware to execute the various acoustic attacks. They primarily worked on week 1 Normal Or Though, week 2 KeyRing 1 and week 4. Also, collaborated with Darshan Singh in the lab and coordinated virtually with Jennifer Maaskant. Additionally, they gained expertise in acoustic analysis using Audacity and algorithm to classify audio.

Jennifer Maaskant contributed the abstract, types of attacks, and methodologies sections of both the qualification and final reports. Throughout the competition, she focused primarily on the week 2 challenges, KeyRing 1 & 2. As team lead, she oversaw the submission of reports, delegated tasks, created PowerPoint and poster templates, and coordinated with the team to prepare for presentations. In addition, Jennifer conducted independent research on previous competition challenges to help the team prepare for this year’s event. She also performed in-depth research on various techniques for executing side channel attacks, using that knowledge to further the group’s progress throughout the competition.

Darshan Singh contributed the mitigation strategies section to both the qualification report and the midterm report. They also did research on how the Arduino Uno was configured in the previous year’s challenge, and worked on how the device would likely be configured in the current year (before the challenge details had been released). Finally, they came up with theoretical attacks on the hardware for the current year based on the parts list that had been released.

IX. CONCLUSION

In conclusion, the 2024 Embedded Security Challenge (ESC) has highlighted the critical vulnerabilities of Cyber-Physical Systems (CPS) to acoustic side-channel attacks. The challenges solved from decoding QR codes through motor noise to cracking simulated safes using acoustic analysis,

underscore the practicality to execute and defend against SCAs. Additionally, this paper explored various SCA, including power attacks, timing attacks, and demonstrating their potential to compromise CPS. Overall, the increasing complexity of CPS has led to a increased attack surface. Especially with programming logic controllers (PLC) and Arduino-based systems which are susceptible to these attacks. As CSP continues to be an integral part of manufacturing processes, showing the need for enhanced security and calling for future research on developing more resilient CPS.

REFERENCES

[1] Splunk, "Cyber-Physical Systems (CPS) Explained," Splunk. [Online]. Available: https://www.splunk.com/en_us/blog/learn/cyber-physical-systems.html. [Accessed: Sep. 14, 2024].

[2] M. Cobb, "What is a side-channel attack?," TechTarget. [Online].Available:https://www.techtarget.com/searchsecurity/definition/side-channel-attack. [Accessed: Sep. 14, 2024].

[3] Oechslin, P. (2003). "Making a Faster Cryptanalytic Time-Memory Trade-Off" (PDF). Advances in Cryptology - CRYPTO 2003. LNCS. Vol. 2729. pp. 617–630. doi:10.1007/978-3-540-45146-4_36

[4] Do A, Ko ST, Htet AT (15 April 2013). "Electromagnetic Side-Channel Analysis on the Intel Atom Processor: A Major Qualifying Project Report" (PDF). Worcester Polytechnic Institute.

[5] D. Agrawal, B. Archambeault, J. R. Rao and P. Rohatgi, "The EM Side-Channel(s)," in Cryptographic Hardware and Embedded Systems - CHES 2002, 2002. [Online]. Available: https://eprint.iacr.org/2004/199/. [Accessed: Sep. 14, 2024].

[6] https://github.com/TrustworthyComputing/csaw_esc_2023/blob/main/Solutions.md

[7] https://www.dhs.gov/sites/default/files/2022-09/22_0902_st_emp_mitigation_best_practices.pdf

[8] M. Stevens et al., "The first collision for full SHA-1," CWI Amsterdam and Google Research. [Online]. Available: https://shattered.io/. [Accessed: Sep. 14, 2024].

[9] X. Wang, D. Feng, X. Lai, H. Yu (2004). "Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD", Cryptology ePrint Archive. [Online]. Available: https://ia.cr/2004/199/. [Accessed: Sep. 15, 2024].

[10] M. Stevens, "Fast Collision Attack on MD5," IACR Cryptology ePrint Archive, 2006. [Online]. Available: https://www.marc-stevens.nl/research/papers/eprint-2006-104-S.pdf. [Accessed: Sep. 14, 2024]

[11] National Institute of Standards and Technology, "Side Channel Attack," Computer Security Resource Center. [Online]. Available: https://csrc.nist.gov/glossary/term/side_channel_attack. [Accessed: Sep. 14, 2024].

[12] Peter Wright. Spycatcher: The candid autobiography of a senior intelligence officer. New York: Viking, 1987.

[13] Harrison, E. Toreini and M. Mehrnezhad, "A Practical Deep Learning-Based Acoustic Side Channel Attack on Keyboards," 2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Delft, Netherlands, 2023, pp. 270-280, doi: 10.1109/EuroSPW59978.2023.00034.

[14] Anindya Maiti, Oscar Armbruster, Murtuza Jadliwala, and Jibo He. Smartwatch-based keystroke inference attacks and context-aware protection mechanisms. In Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, pages 795–806, 2016.

[15] M. A. Al Faruque, S. R. Chhetri, A. Canedo and J. Wan, "Acoustic Side-Channel Attacks on Additive Manufacturing Systems," 2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS), Vienna, Austria, 2016, pp. 1-10, doi: 10.1109/ICCPS.2016.7479068.

[16] Yarom, Yuval, et al. “CacheBleed: A Timing Attack on OpenSSL Constant-Time RSA.” Journal of Cryptographic Engineering, vol. 7, no. 2, 11 Feb. 2017, pp. 99–112, https://doi.org/10.1007/s13389-017-0152-y.

[17] Yun, Cathie. “Adventures with RSA Blind Signing - Cathie Yun - Medium.” *Medium*, 25 Feb. 2021, cathieyun.medium.com/adventures-with-rsa-blind-signing-397035585121.

[18] Genkin, Daniel, et al. “Get Your Hands off My Laptop: Physical Side-Channel Key-Extraction Attacks on PCs.” *Journal of Cryptographic Engineering*, vol. 5, no. 2, 6 May 2015, pp. 95–112, https://doi.org/10.1007/s13389-015-0100-7. Accessed 7 Mar. 2020.