Leveraging EM Side-Channels for Recognizing Components on a Motherboard

Frank T. Werner, *Student Member, IEEE*, Baki B. Yilmaz *Student Member, IEEE*, Milos Prvulovic, *Senior Member, IEEE*, and Alenka Zajić, *Senior Member, IEEE*

***Abstract*—This paper proposes leveraging EM side-channels to recognize/authenticate electronic components integrated onto a motherboard. By focusing on components on a motherboard, our method provides an opportunity to authenticate devices assembled by third parties. This method identifies components based on the modulated signals emanated while they are excited in a controlled manner. When testing an unknown component, the spectrum is compared to previously recorded training signatures. To improve efficiency, the size of the spectrum is reduced by projecting it into a vector space generated from training signatures. The identity of the tested component is then determined using a k-Nearest Neighbors algorithm. This method successfully classified memory, processor, and Ethernet transceiver components integrated on seven types of Internet-of-Things devices. Since manufacturers commonly use the same components in multiple designs, cross-type testing of motherboards is conducted. Collecting the training signatures on one motherboard and testing components from different motherboards speeds up the process and decreases the cost. Using measurements taken while exciting the components for 1 s, our method achieves a classification accuracy greater than 96% across all components tested. These results demonstrate that this method can recognize components based on their emanations, even if the components are integrated onto completely different motherboards.**

***Index Terms*—EM side-channels, component recognition/authentication, supply chain verification.**

I. Introduction

COMPUTER systems and other electronic devices are typically assembled by a system integrator. These systems contain one or more printed circuit boards (PCBs) that have been procured (sourced) from various suppliers. These PCBs are themselves produced by board-level integrators, and typically contain a number of integrated circuits (ICs) that are also sourced from various suppliers. The procurement ecosystem for both ICs and PCBs is sophisticated, meaning the actual manufacturer of an IC very rarely acts as a direct supplier to a PCB integrator, and the PCB integrator is very rarely a direct supplier to the system integrator. However, for both the system integrator and the end user of the system, it is important to know which actual devices (both IC- and PCB-level) are present in the system. Different devices, even when they are functionally compatible, differ in other properties, such as the level of trust its manufacturer enjoys, the level of reliability and environmental tolerance the device can be expected to provide, the inter-operability issues with specific software and with other devices, and the bugs/vulnerabilities that must be taken into account to ensure that the system functions correctly and securely.

Unfortunately, the complex supply chain for both ICs and PCBs makes it difficult to avoid counterfeits [1], which are thought to represent around 1% of all semiconductor sales [2] and cost legitimate component manufacturers approximately $\$100$ billion in lost sales [3]. Furthermore, even in the absence of malicious intent, one legitimate device can be (and often is) substituted with another legitimate device that the PCB manufacturer may consider to be equivalent. However, some of the properties of these devices may differ, especially when it comes to inter-operability, bugs, and vulnerabilities. To overcome this problem, it is very important to correctly recognize/authenticate components on a PCB or in a system, so that the appropriate software patches and workarounds can be applied, and so that tracking and mitigation of reliability and inter-operability issues can be correctly implemented.

Currently, industry relies on several different methods for recognition/authentication of electronic parts. The most commonly used can be broken into two types: physical inspection and electrical inspection [4]. Physical inspection includes visually examining the inside and outside of the components and analyzing their material composition. Electrical inspection includes testing the components' electrical characteristics, performance, and durability through burn-in tests [4]. However, the effectiveness of these methods can vary based on the type of counterfeit, level of intrusiveness during testing, cost, time, and other conditions [1]. Reliable, nondestructive approaches that allow for easy, precise, and cost-effective recognition/authentication of electronic components are needed.

The Supply Chain Hardware Integrity for Electronics Defense (SHIELD) is an example of a authentication method developed to overcome many of these challenges [5]. SHIELD relies on a passive sensor called a "dielet" for authentication. During assembly, the dielet is inserted into the component's package and records unauthorized attempts to physical access or modify the component. The information in the dielet can then be accessed wirelessly during authentication. While this method may be effective for detecting multiple types of counterfeits, it requires additional hardware to be added to the component, increasing its cost.

A promising alternative is to rely on electromagnetic (EM)

This work has been supported, in part, by DARPA LADS contract FA8650-16-C-7620 and ONR grant N00014-19-1-2287. The views and finding in this paper are those of the authors and do not reflect the views of DARPA or ONR.

Frank T. Werner, Baki B. Yilmaz, and Alenka Zajić are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA,
Milos Prvulovic is with the School of Computer Science, Georgia Institute of Technology, Atlanta, GA 30332, USA.

side-channels, which are unintentional emanations created as a consequence of transistors switching on a component while it is performing computations [6], [7]. Other types of side-channels besides EM include power [8], acoustic [9], temperature [10], backscattering [11], and timing [12]. While the most well-known application of EM side-channels is for stealing confidential information [13]-[16], they can also be used in more benign applications such as malware detection [17]-[19], hardware Trojan detection [20], [21], and counterfeit detection [22], [23]. EM side-channels have also been used for identifying devices such as automobiles, desktops, phone chargers, cell phones, toys, and microcontrollers [24]-[31]. *However, all these methods focus on identifying PCB-level devices, and no attempts have been made to recognize/authenticate components on a PCB within the device.*

To address this problem, this paper proposes to leverage EM side-channels to recognize/authenticate components integrated onto a motherboard. By focusing on components on a motherboard, our method provides an opportunity for designers and manufacturers to authenticate devices assembled by third parties. The purpose is to detect counterfeit ICs based on changes in the EM emanations that comprise the EM side-channel. This method is intended for detecting types of counterfeiting where the physical design of the component is altered. Examples include cases where the intended IC has been replaced with a reverse engineered copy or with a lower quality component and cases where the design has been tampered with [1].

The proposed method authenticates components based on the modulated signals emanated while the component is active. These signals are generated by exciting the component in a controlled and repeatable manner. While the component is being excited, the emanated spectrum is recorded. During training, the emanations from several examples of each component are recorded. In this situation, the components' identities are known and the components are assumed to not have been tampered with. The spectrums recorded during the training are used as signatures for the components of interest. When testing an unknown component, its spectrum is recorded and then compared to the previously recorded training signatures. However, the spectrums are not compared to directly. Instead, to improve the efficiency, the size of the spectrum is first reduced by projecting it into a vector space generated from the training signatures. The identity of the tested component is then determined using a k-Nearest Neighbors (k-NN) algorithm [32].

More complicated and expensive ICs, such as memories and processors, are the focus of this method, since they are some of the most important components the device. Counterfeits of passive components on the motherboard, such as capacitors or inductors, cannot be detected using this method; however, they are less critical to the device. Unlike other authentication methods, our method requires no additional hardware, nor does it damage the component during testing, hence reducing its cost. At the same time, the equipment for detecting counterfeits is always under the control of the user, unlike for SHIELD, where the dielet itself could be tampered with.

The proposed method has successfully classified external memory, processor, and Ethernet transceiver components integrated on seven types of Internet-of-Things (IoT) devices. Nine to ten different instances of each device are used in the experiments. Cross-type testing of boards is conducted as well. Since manufacturers commonly use the same components in multiple designs, being able to collect the training signatures on one motherboard and test components from different motherboards significantly speeds up the process and decreases the cost. Using the measurements taken while the components are excited for 1 s, our method achieves a classification accuracy greater than 96% for all tested components. These results demonstrate that this method can recognize components based on their EM emanations even if they are integrated on different a type of motherboard.

The rest of this paper is organized as follows. Section II describes the approach used to excite components and maximize the presence of EM side-channel features. Section III describes how the measurements are processed to improve the efficiency of the identification. Section IV details the process for training on and testing components. Section V describes the experiments performed to evaluate the proposed method. Finally, Section VI presents the conclusions.

II. Signals Carrying Information About Device Side-Channel Signatures

This section discusses what spectral features are found to be relevant for component recognition and how we excite electronic components in order to maximize the presence of the EM side-channel features.

A. Spectral Features

Our method relies specifically on the modulated signals emanated by the component for identification. These are some of the strongest signals available in the side-channel [33]. They are caused by the device's program activity unintentionally modulating periodic signals, such as synchronization clocks, already present in a device [33]. Program activity results in the superposition of a time-varying current on the traces inside and connecting the components used for the execution. The magnitude of the emanations depends on the change in power when executing the activity, while modulation frequency is related to the time it takes to execute the repetitive behavior. While multiple types of modulation can occur in a device, this work relies only on amplitude modulated (AM) signals generated by a component for identification [34].

We have observed that any change in the component or program activity affects the properties of the emitted AM signal. For example, the shape and spread of the sidebands in the frequency spectrum are related to the time it takes to execute parts of the program activity. If the execution time varies, the sideband's shape will contort and spread in frequency. By keeping the program activity consistent for all tests, the spectral features of the emanations can be used as a signature for identifying the components. Experimentally we have determined that these features include: 1) the overall modulation frequency, 2) the sideband shape and spread, 3) the relative strength of the sideband's fundamental frequency and

the higher harmonics, 4) the carrier frequency, and 5) the carrier spread. Fig. 2 in the following subsection provides an example of a signal with these properties highlighted.

B. Device Excitation

When selecting spectral features to use for identification, the device needs to be in a known and repeatable state. In this state, the component of interest needs to be active; otherwise, there will be nothing in the spectrum to use for identification. For example, leaving the device in standby is an obvious option for a measurement state. However, the device will not be very active, making it difficult to find useful features for identification (assuming the component is active at all).

Even if the component is being excited, it can still be difficult to locate useful spectral features. Depending on the program activity, the spectrum can change rapidly in time. As a result, locating useful spectral features, in both time and frequency, can be challenging. Selecting a measurement state where the device's spectrum is relatively constant makes identification easier.

To address these challenges, the excitation program described in [35] is used to excite the device in during testing. Using the excitation program, controllable emanations are generated by executing an alternating pattern of two instructions, such as addition, subtraction, load, and store. These instructions are defined by the processor's architecture and how they are executed is independent of the devices' operating systems (OS). Generally, the generated emanations are much stronger than other activities in the spectrum, minimizing influence of these other activities. Furthermore, this approach makes it possible to excite different parts of the device. As demonstrated in [36], different instructions will excite different components (such as the processor and RAM), depending on what is being used for execution. This effect limits the influence other components have during the measurements.

The excitation program can be easily implemented on a variety of devices, making it ideal for authenticating components integrated on multiple types of motherboards. It has already been implemented on several different types of laptops, desktops, cell phones, computer boards, and FPGAs [35]-[36]. In these experiments, the devices' OS included several versions of Windows, multiple different implementations of Linux, and the embedded operating systems unique to the FPGA.

An example of the excitation program is shown in Fig. 1 [36]. In this example, the first program activity is X, and the second is Y. The program is comprised of two smaller loops contained in an outer loop. The first inner loop repeatedly executes X, while the second repeatedly executes Y. The variables "$inst\_x\_count$" and "$inst\_y\_count$" define the number of times X and Y are executed in their respective loops. The execution time of the outer loop is equal to one period, T_{alt}, of the excitation signal. The alternating frequency, f_{alt}, is equal to $\frac{1}{T_{alt}}$. The excitation program generates a signal at f_{alt} and its harmonics ($2f_{alt}$, $3f_{alt}$, ...). The frequency of the emanations can be tuned by changing the number of times activities X and Y are executed.

```
1  while(true){
2    // Execute the X activity
3    for(i=0;i<inst_x_count;i++){
4      ptr1=(ptr1&~mask1)|((ptr1+offset)&mask1);
5      // The X-instruction, e.g., a load from L2
6      value=*ptr1;
7    }
8    // Execute the Y activity
9    for(i=0;i<inst_y_count;i++){
10     ptr2=(ptr2&~mask2)|((ptr2+offset)&mask2);
11     // The Y-instruction, e.g a store from L2
12     *ptr2=value;
13   }
14 }
```

Fig. 1. Pseudo-code to generate the X/Y alternation activity [36].

The duty cycle of the excitation is based on the amount of time spent executing each instruction. For all of the experiments in Section V, the duty cycle is set to 50%; therefore, the modulating baseband signal should be a square wave. While a square wave should have only odd harmonics, there will likely be weak even harmonics present in the generated spectrum. Since it takes a fixed amount of time to execute each instruction, it is difficult to tune the waveform to have a precise duty cycle. This distortion can be used as a factor in identifying the component since the execution time of each instruction is dependent on the components used for the execution.

An example of the spectrum (in dB) generated using the excitation program is shown in Fig. 2. To normalize the signal, it has been divided by its mean. In the figure, the carrier and the modulated sidebands caused by the excitation program are labeled. The excitation has a fundamental frequency of 10 kHz with higher harmonics at ± 20 and ± 30 kHz from the carrier. Since the duty cycle is set to 50%, the odd harmonics are much stronger than the even. However, the presence of the weak even harmonics at ± 20 kHz indicates the actual duty cycle is not exactly 50%. The shape and spread of the sidebands are caused by variations in the execution time of the excitation program, while the spread of the carrier is caused by the instability of its source. Finally, the power of the sidebands relative to each other and to the carrier are based on three factors: the physical properties of the component, the excitation program, and the location of the measurement probe. *Keeping the excitation program and probe position fixed allows us to observe the physical properties of the components.*

III. Signal Compression and Processing

After the component has been excited and its emanations have been recorded, the measured signal is processed to make it easier to locate important features for identification. To eliminate the influence of time variations and noise on the measurements and to improve the efficiency of identification, an updated version of the approach described in [39] is used. This new approach is described below.

During a test, the EM emanations from the excited component are recorded for a period of time, T. The number of samples, M, is equal to the measurement time multiplied by the sample frequency, f_{s}. To better emphasize the elements of the signal used for identification, the signal is converted to the frequency domain. However, instead of converting the entire signal at once, the measurement is first broken into several

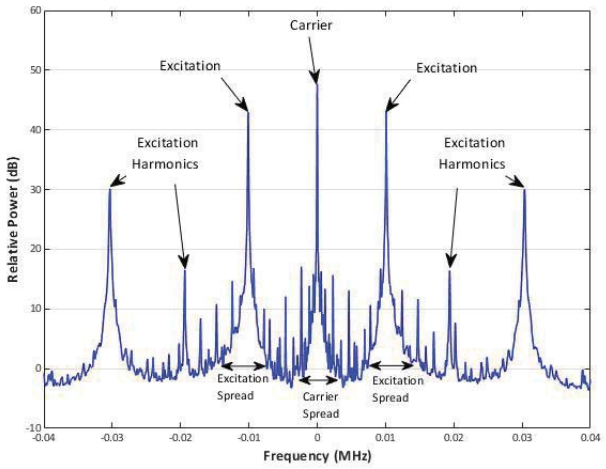


Fig. 2. Example of the spectrum produced by the excitation program.

segments, N_{R}. A short-time discrete Fourier transform (STFT) is then applied to each segment. A flattop window is used to improve the accuracy of the relative amplitudes of different frequency components.

Each STFT operation for the kth frequency component is calculated by

$$Y_h[k] = \sum_{n=1}^{N} y[n + (h-1)N_{\mathrm{s}}]w[n-k]\exp(-\mathrm{i}2\pi kn/N) \quad (1)$$

where y is the original measurement, h is the STFT operation number, w is the flattop window function, N_{s} is the number of nonoverlapping samples, and N is the window size for the STFT operations. The nonoverlap time is the number of samples, N_{s}, shifted between STFT segments.

The number of elements in each segment, N, affects the resolution of the window. The resolution needs to be high enough for features of the spectrum to be distinguishable. However, if the size of N is too large, it will increase the measurement and processing times. After M, N, and N_{s} have been defined, the number of STFT operations performed for the data can be calculated by

$$N_{\mathrm{R}} = \mathrm{floor}\left(\frac{M-N}{N_{\mathrm{s}}} - 1\right). \quad (2)$$

To reduce the impact of noise and other time variations in the signal on the spectrum, these N_{R} STFT operations are averaged together as

$$\overline{\mathrm{Y}}[k] = \frac{1}{N_{\mathrm{R}}}\sum_{h=1}^{N_{\mathrm{R}}} |Y_h[k]| \quad (3)$$

where $\overline{\mathrm{Y}}$ is a row vector containing the averaged frequency magnitudes. Here, to eliminate the impact of the starting time on the data, the phase is removed by taking the magnitudes of the STFT operations. Afterwards, the strongest component in the spectrum is downconverted to 0 Hz. This component is assumed to be the carrier. Since the exact frequency of the carrier is influenced by factors such as manufacturing variability and temperature, the carrier frequencies of two identical components on separate devices will be slightly different. Shifting the carrier to the center of the spectrum ensures that this difference does not influence the identification. Afterwards, a band-pass filter is applied to the downconverted signal by removing 10% of the bandwidth. This filtering ensures that the carrier is located at the center of the spectrum.

Next the measurements are converted from a linear scale to a decibels scale (dB) using

$$\overline{\mathrm{Y}}_{h_{\mathrm{dB}}}[k] = 20\log_{10}(\overline{\mathrm{Y}}_h[k]). \quad (4)$$

This conversion reduces the influence of the strong signal components, while increasing the influence of weaker components in the spectrum. Since the modulation is not intentional, the carrier tends to be significantly stronger than the sideband components, usually tens to hundreds of times stronger. Without this conversion, the carrier has a disproportionate influence on the data. Afterwards, the measurements are standardized by subtracting their mean and dividing by their standard deviation.

Once all the measurements are processed, they are combined in the HxN matrix

$$\mathbf{Y} = \begin{bmatrix} \overline{\mathrm{Y}}_{1_{\mathrm{dB}}} \\ \overline{\mathrm{Y}}_{2_{\mathrm{dB}}} \\ \vdots \\ \overline{\mathrm{Y}}_{H_{\mathrm{dB}}} \end{bmatrix}, \quad (5)$$

where H is the number of measurements. In other words, each row represents the averaged frequency components of a measurement. However, the matrix generally contains redundant information because H and N are not equal. To reduce the size of the data and improve efficiency, singular value decomposition (SVD) is applied to the data matrix. As a result, the data matrix is decomposed into the following form:

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (6)$$

where $\mathbf{U}$ is the left singular vectors matrix, $\mathbf{\Sigma}$ is the singular values matrix, and $\mathbf{V}^T$ is the transpose of the right singular vectors matrix. The measurement data is then projected into a new vector space, $\boldsymbol{Z} \in \Re^{H\times K}$, by

$$\boldsymbol{Z} = \mathbf{Y}\mathbf{V}_K, \quad (7)$$

where $\mathbf{V}_K$ has been reduced to $K \times K$ matrix. The size of K is significantly smaller than the length of the original measurements and corresponds to the K largest singular values in $\mathbf{\Sigma}$. These K vectors represent the signal components of the data matrix that contains the directions corresponding to largest K singular values. As a result, the original $H \times N$ dimensional feature space of the measurements has been reduced to $H \times K$ dimensions, without much loss in information about the data.

IV. Training and Testing Process

The identification process can be broken into training and testing phases. In the training phase, EM signatures from

example components are recorded and processed. Here, the identities of the components are already known. If the training components are used on multiple types of devices, only one of the devices needs to be used for training. Each training component is excited using the excitation program, and its emanations are recorded for T seconds. The modulation frequency used for the excitation program is set beforehand and kept constant for all measurements. The carrier frequency of the component is located using either a method such as the ones detailed in [33] and [34] or manually. The measurements are processed into averaged STFTs and stored in a matrix. If multiple types of components are tested, such as memory and processor, the signatures can be divided into multiple matrices based on the component's function or the carrier frequency. The training matrix or matrices are then decomposed using SVD. The first K columns from the resulting matrix $\mathbf{V}$ are selected. Next, the matrix $\mathbf{V}_K$ is used to project the training data into the new vector space. In other words, assuming the projected training data set is $\mathcal{Y}$, we generate a model ($\mathcal{Y}$, $\mathbf{V}$) that is used in the testing phase.

In the testing phase, the EM emanations generated by one or more unknown components are recorded. As in the training phase, the tested component is excited using the excitation program. The measurements are then processed into averaged STFTs. The resulting signatures are projected into a new vector space using the matrix created from the training data. Afterwards a k-NN algorithm is applied to the projected training and testing data by using the model ($\mathcal{Y}$, $\mathbf{V}$). The k-NN algorithm determines the identity of the tested component based on the standardized Euclidean distance between the projected measurements and the training measurements.

Although k-NN is a fairly simple type of clustering method, it is a practical tool for classifying components. As the experiments in the following section demonstrate, it can accurately determine the identities of several types of components.

V. Experimental Validation

To demonstrate its effectiveness, the new identification method is applied on components from seven types of IoT devices from Olimex and two FPGA developement boards. This method is not limited to such devices; we use them here only as examples. All the IoT devices are from the same manufacturer (Olimex) because it presents the most difficult case for identifying components. As demonstrated in Subsection V-C, contrasts between devices from different manufacturers are even larger, making them easier to identify.

For the experiments, the external memory, processor, and Ethernet transceiver components from each device are tested. In the following subsections, the measurement setup, the tested components, and the experimental results are discussed. The experimental results are broken into four subsections. Subsection V-C demonstrates the effect of projecting the measurement data in the new feature space, while the other three subsections (V-D to V-F) describe the results of testing on different types of components. The measurement settings described in the measurement setup applies to these three subsections. Subsection V-C uses a simpler combination of settings.

A. Measurement Setup

Fig. 3 shows the measurement setup used for the experiments. A small hand-made circular coil probe with a 1 mm radius is used to measure the magnetic field emanated by the component being tested. This probe was selected so that its small size would limit the influence of emanations from other nearby components on measurements. The probe is connected to a Keysight M9391A PXIe Vector Signal Analyzer (VSA) for recording the signal. During the measurements, the probe is placed directly on top of the monitored component, where the emanations are the strongest. To ensure consistency between measurements, the probe is positioned using the EM Probe Station's motorized XYZ table from Riscure [40]. The table is controlled through a USB port using a laptop. The laptop is also used to control the device under test (DUT) and the VSA through their Ethernet ports. In cases where the DUT does not have an Ethernet port, a USB-to-Ethernet adapter is used. A diagram of the test setup is shown in Fig. 4.

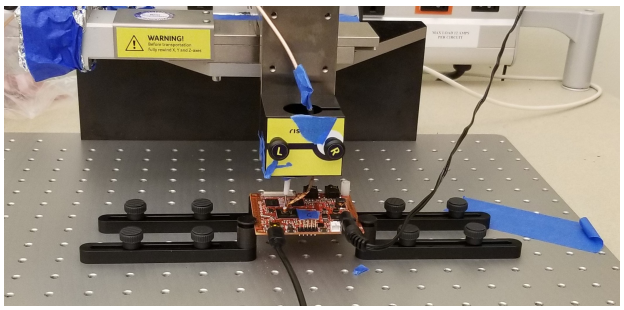


Fig. 3. Measurement setup used for the experiments.

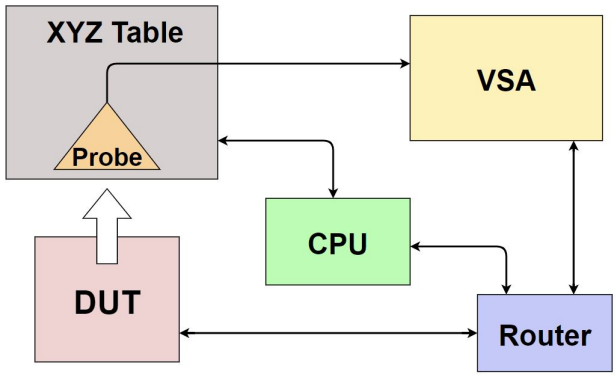


Fig. 4. Diagram of the measurement setup.

During a measurement, all the components are excited using the program outlined in Subsection II-B. The program generates a 10 kHz excitation signal with a 50% duty cycle by executing an alternating pattern of addition and load instructions. When measuring the external memory components, the array size of the load instruction is set to be much larger (8.4 MB) than the processor's cache to ensure that the external memory is active during the measurements. When exciting the processors, the array size of the load instructions is kept small (8.2 kB) to ensure that the load execution is mostly confined to the processor's cache. This small array size minimizes the influence the external memory has on the signal. When testing

multicore processors, the excitation program is executed only on the first core, while the rest are left idle.

While a component is excited, its emanations are recorded for 1 s (which is T in Section III). In Subsections V-D to V-F, each of the seven types of IoT devices had 10 individual units; however, one unit (an A33-MAIN) is removed from the results since it is damaged. Therefore, a total of 69 boards are tested. Furthermore, cross-type testing is conducted because some of the device types have the same components. In these situations, measurements from only one type of devices are used for training. Similarly, in situations where the same component is used multiple times on the same device, measurements from only one instance of the component are used for training.

To account for the limited number of boards, a k-fold cross-validation scheme with five-folds is run for 10 000 iterations. Such schemes are commonly used in cases where the sample size is small since their results have a relatively low bias and variance compared to other cross-validation approaches [41]. The accuracy of correctly identifying each type of component and the overall accuracy of correctly identifying all the components are calculated by averaging the results for each iteration.

Several of the component measurements have interrupts caused by their device's operating systems [42]. To improve consistency and ensure the spectrum is primarily a result of the excitation program, these interrupts are removed from the measurements. The measurement bandwidth is 220 kHz (reduced to 200 kHz after processing). A small bandwidth is used to reduce the amount of interference present in the measurement. This interference may help distinguish different components from each other but can also make the same components on different device types appear dissimilar.

During processing, each measurement is processed into 43, N_{R}, STFTs with a length, N, of 20 480 before being averaged. The number of non-overlapping samples, N_{s}, is 4 000, which corresponds to 20 ms. After the measurements are processed, the training measurements are used to generate a new vector space for evaluation. Afterwards, the k-NN algorithm is applied to the testing data by using the model ($\mathcal{Y}$, $\mathbf{V}$). For the following sections, only the first four dimensions of the new vector space are used for evaluation, i.e. $K = 4$. As mentioned previously, the k-NN algorithm classifies a measurement based on the standardized Euclidean distance between the measurement and each training signature. The differences between the four coordinates of the test and training points are scaled by dividing by 1, 1, 2, and 3, respectively, for the memory and processor and by 1, 1, 3, and 3 for the Ethernet transceivers. The number of dimensions and the scaling factors can be tuned to improve the classification accuracy for a specific set of components.

B. Test Devices

In the experiments, seven types of IoT devices from Olimex are tested. These devices are the A10-OLinuXino-LIME [43], A13-OLinuXino [44], A13-OLinuXino-MICRO [45], A20-OLinuXino-LIME [46], A20-OLinuXino-LIME2 [47], A20-OLinuXino-MICRO [48], and A33-OLinuXino [49]. For simplicity, these devices will be referred to as A10-LIME, A13-MAIN, A13-MICRO, A20-LIME1, A20-LIME2, A20-MICRO, and A33-MAIN for the rest of this work. All the devices run Linux OS provided by Olimex. Instead of being installed on the device itself, the OS are saved to SD cards. The only change made to the devices was installing the excitation program.

Pictures of tested IoT devices are shown in Fig. 5. All the IoT devices are part of Olimex's OLinuXino open source hardware product line. They are convenient options for these experiments since Olimex has provided detailed information about each device (such as the schematic, parts list, and PCB layout) on their website [50].

In Subsection V-C, two extra memory components (MEM5 and MEM6) are tested to demonstrate the impact of projecting the measurements into the new feature space. These extra components are integrated into DE0-CV Cyclone V [51] and DE1 Cyclone II development boards [52]. Pictures of the devices are shown in Fig. 6. These components and their motherboards are significantly different from the Olimex devices, both in functionality and physical properties. The differences result in the projected data from the development boards being easily distinguishable from the IoT measurements. Therefore, these components are not included in later subsections. Correctly classifying similar, yet physically different components is more challenging, especially if the components are integrated onto similar motherboards from the same manufacturers.

The reason motherboards from the same manufacturer increase the difficulty is that manufacturers commonly use the same components and PCB layouts in multiple designs to save time and money. These similarities can influence the parts of the spectrum not related to the component of interest. The A10-LIME and A20-LIME1 are examples of reused PCB layouts. The PCB for the A10-LIME is an older revision of the PCB for the A20-LIME1. In this situation, the traces on both motherboards will have similar emanation properties since they are almost identical in shape and composition. If one motherboard is made by a different manufacturer, the emanation properties change since the physical configurations of the traces and the material properties of the PCB would change. However, in either case, the signal properties of the emanations from the traces and the components still depend on the component and the program activity.

The following experiments focus on identifying the external memory, processor, and Ethernet transceiver components present on each device. Some of these components are not present on all the devices, while, in other cases, some devices use the same components as others. A complete list of the components, the devices they are present on, and the measurement frequency are provided in Table I. For simplicity, the components will be referred to by their label in the table. More information about the devices and their components can be found on the component manufacturer's websites (referenced in the table).

C. Measurement Projection

Before discussing the method's performance, the impact of projecting the measurement data into the new feature space

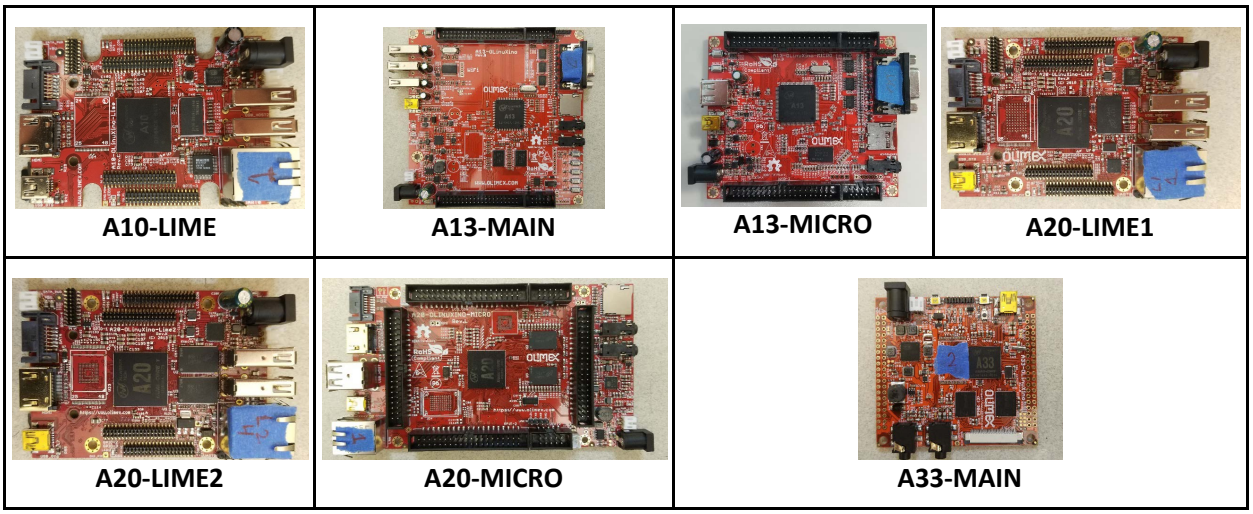


Fig. 5. Pictures of the seven IoT devices (not to scale).

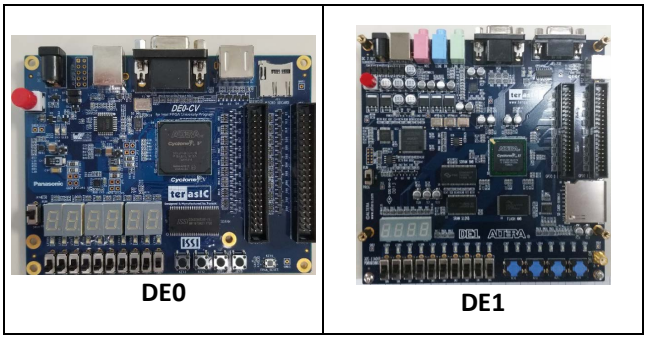


Fig. 6. Pictures of the DE0-CV Cyclone V and DE1 Cyclone II development boards (not to scale).

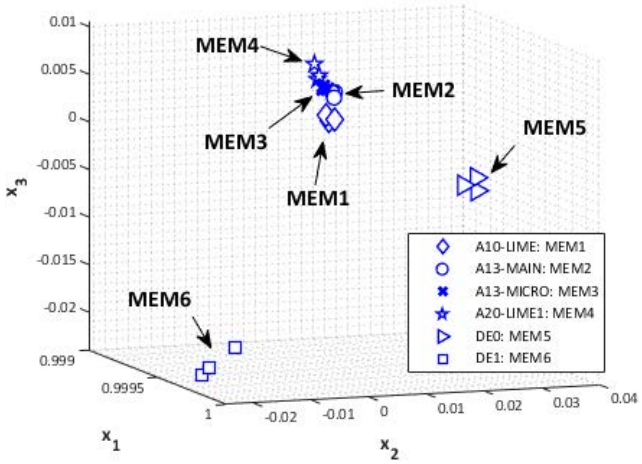


Fig. 7. Example of the memory measurements projected into the new feature space.

needs to be explored. As an example, three measurements from all six types of memory components are projected into a new 3D feature space (shown in Fig. 7). All measurements are recorded for 0.2 s with a bandwidth of 1 MHz and an excitation frequency of 100 kHz. For simplicity, the example measurements are used for generating the feature space before being projected into it.

In Fig. 7, the points from MEM5 and MEM6 are relatively far away from the other components. Their isolation is the result of their measurements having significantly different spectral features. When the measurements are projected, the differences in the spectral features translate into different coordinates in the feature space. These differences are the result of the first four memory components being radically different from the last two. While all six are SDRAM, the first four are DDR3 with operating frequencies greater than 300 MHz, while the last two are SDR (single data rate) with operating frequency below 200 MHz.

On the other hand, points from MEM1 through MEM4 are clustered close enough together that they are difficult to visually distinguish from one another. The reason for this clustering is that the components are similar in functionality to one another. Furthermore, the fact the motherboards are from the same manufacturer likely helped their similarities. However, the closer the projected points from different types of components are to one another, the greater the risk of them being misclassified.

More detail about the relationships between the measurements can be gained by comparing the separation distances between the projected data. The average distances between data from two different types of components and the average distance between data from the same type of component can be calculated. For simplicity, the distance between points from two different types of components is called the *class-distance*, and the distance between points belonging to the same type of component is called the *self-distance*. The class-distance is the result of the measurements from the different components having different spectral features. The larger the distance, the greater the difference. The self-distance is caused by differences in the spectrums measured from two of the same type. These differences can be the result of changes in how the measurements are taken (such as probe type and probe position), changes in the program execution over

TABLE I
LIST OF TESTED COMPONENTS

Label	IC Name	IC Type	Devices	Carrier Frequency (MHz)	Source
MEM1	K4B4G1646Q-HYK0	4Gb DDR3 SDRAM	A10-LIME	384	[53]
MEM2	H5TQ2G83FFR	2Gb DDR3 SDRAM	A13-MAIN U1 and U2	408	[54]
MEM3	H5TQ2G63BFR	2Gb DDR3 SDRAM	A13-MICRO	408	[55]
MEM4	MT41K256M16HA-125:E	4Gb DDR3 SDRAM	A20-LIME1, A20-LIME2 U2 and U3, A20-MICRO U2 and U3	384	[56]
MEM5	IS42S16320D-7TL	512Mb SDR SDRAM	DE0	100	[57]
MEM6	A2V64S40CTP-G7	64Mb SDR SDRAM	DE1	50	[58]
PROC1A	Allwinner A10 (Cortex-A8)	SoC (Processor)	A10-LIME	1008	[59]
PROC1B	Allwinner A13 (Cortex-A8)	SoC (Processor)	A13-MAIN, A13-MICRO	1008	[60]
PROC2A	Allwinner A20 (Cortex-A7)	SoC (Processor)	A20-LIME1, A20-LIME2, A20-MICRO	960	[61]
PROC2B	Allwinner A33 (Cortex-A7)	SoC (Processor)	A33-MAIN	960	[62]
ETH1	RTL8201CP	Ethernet Transceiver	A10-LIME	25	[63]
ETH2	LAN8710A-EZC-TR	Ethernet Transceiver	A20-LIME1, A20-MICRO	25	[64]
ETH3	KSZ9031RNXCC-TR	Ethernet Transceiver	A20-LIME2	25	[65]

time, and manufacturing variation. By the design, the test process described in Section IV minimizes the first two factors. However, the influence of manufacturing variation cannot be removed.

These manufacturing variations are the cause of random fluctuation during the manufacturing process of an IC. These fluctuations result in small physical distinctions between individual ICs of the same type and are the basis of physically unclonable functions (PUFs) [66]. These small distinctions can impact the EM emanations generated by the IC, causing slight disparities in the measurements taken on individuals of the same type. While the impact of manufacturing variation cannot be removed, the likelihood of the manufacturing variation causing a misclassification can be evaluated by comparing the class-distances and self-distances of the data. If the self-distance for a component is much smaller than the component's class-distances, it can be concluded that the effect of the manufacturing variation is outweighed by the dissimilarities in the spectral features between the types of components. Therefore, the chance of manufacturing variation causing misclassification is small.

The average distances between the measurements for each type of memory are shown below in Table II. The diagonal values (in bold) are the self-distances for each component, while the rest are the average class-distances between different types of components. For readability, the values here and in later sections have been multiplied 100. The values themselves are unitless and their only significance is their size relative to one another.

Reflecting the results in Fig. 7, the class-distances between the first four memories and MEM5/MEM6 are significantly higher than the class-distances between the first four memory components only. For example, the class-distance between MEM1 and MEM4 is 9.7 while the class-distance between MEM1 and MEM6 is 59.9.

Furthermore, the table demonstrates that the class-distances between each component are larger than self-distances. This indicates that the impact of the manufacturing variation is

TABLE II
AVERAGE DISTANCES BETWEEN SELECT MEMORY COMPONENTS

	MEM1	MEM2	MEM3	MEM4	MEM5	MEM6
MEM1	**1.8**	5.6	7.3	9.7	80.4	59.9
MEM2	5.6	**1.1**	3.0	4.8	82.1	64.9
MEM3	7.3	3.0	**1.2**	2.7	85.1	65.0
MEM4	9.7	4.8	2.7	**1.9**	86.1	67.4
MEM5	80.4	82.1	85.1	86.1	**15.4**	107.8
MEM6	59.9	64.9	65.0	67.4	107.8	**6.1**

outweighed by the differences between different types of components. Therefore, the risk of the manufacturing variation between the tested components causing misclassification is small.

D. Recognition of Memory Components

Next the devices with the first four memory components are evaluated. Only one MEM1, MEM3, and MEM4 component is integrated on the A10-MAIN and A13-MICRO, and A20-LIME1, respectively. However, A13-MAIN has two MEM2s, and the A20-LIME2 and A20-MICRO have two MEM4s. For devices with more than one of the same type of memory component, each component is measured and evaluated separately. The external memory from the A33-MAIN is not included since it uses a spread spectrum memory clock.

Fig. 8 shows a comparison between the spectrums measured from MEM1 on a A10-LIME, MEM2 on a A13-MICRO, MEM3 on a A13-MAIN, and MEM4 on a A20-LIME1. For example, MEM1's carrier has a much stronger and wider spread than the other three components. Furthermore, the relative strength of the odd sidebands for MEM1 is lower than the other memory components, while the even harmonics are much stronger.

At the same time, the signatures for MEM2 and MEM3 are similar, an unsurprising result given that the components are from the same product line. (The part numbers for MEM2 and MEM3 are H5TQ2G83FFR and H5TQ2G63BFR respectively). However, noticeable differences can be identified between the two components. For example, the harmonics of the MEM3 are more spread out compared to those of MEM2.

The sidebands for MEM4 are the strongest among the memory. Furthermore, the spread of the sidebands for MEM4 is much larger than the spread of the other components. This spread is especially noticeable at the first harmonics, where it is nearly 10 kHz.

Example measurements from A10-LIME, A13-MAIN, A13-MICRO, and A20-LIME1 are used for training. These devices are selected because they have the cleanest spectrums. Since there are two instances of MEM2 on A13-MAIN, only the components designated U1 on the motherboard are used for training. While all the A20 devices use the same memory components, only A20-LIME1 are used for training. The overall classification accuracy for the memory components after cross-validation is 100%. Since there were no classification errors, a confusion matrix for the results is not provided. The algorithm had no difficulty correctly classifying all the memory components, even MEM2 and MEM3, since the distinguishing spectral features are prominent and unique for each memory component.

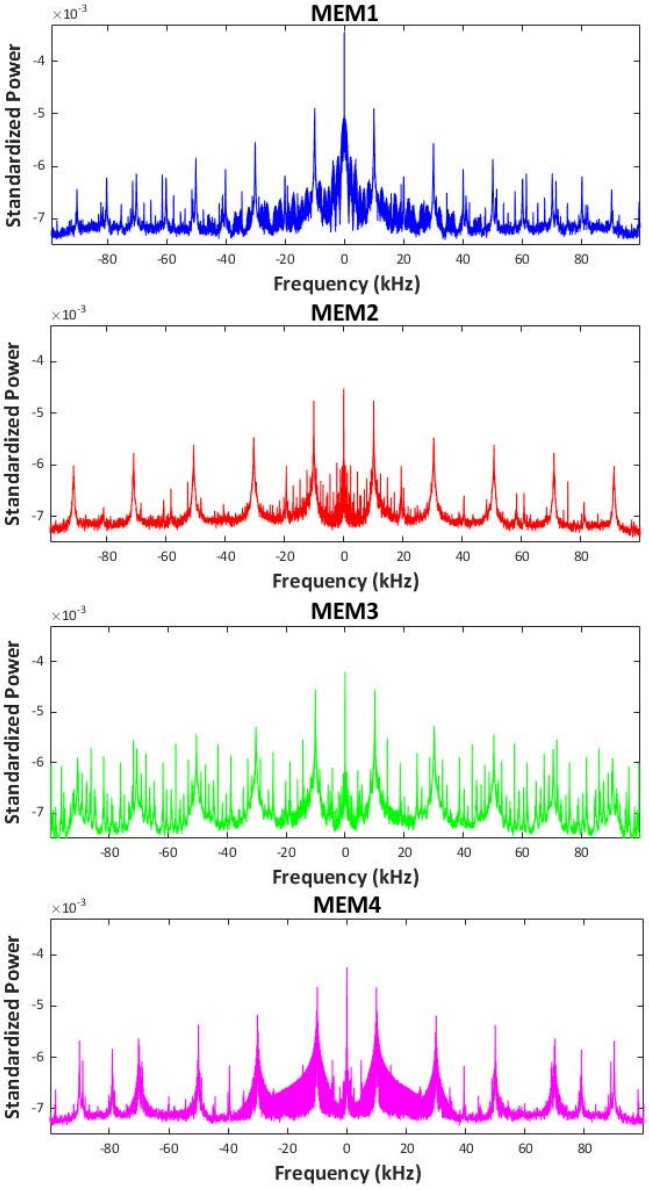


Fig. 8. Comparison of the EM signatures from MEM1 (top in blue), MEM2 (second from the top in red), MEM3 (third from the top in green), and MEM4 (bottom in magenta).

TABLE III
AVERAGE DISTANCES BETWEEN MEMORY COMPONENTS

	MEM1	MEM2	MEM3	MEM3
MEM1	**1.0**	34.7	50.6	32.1
MEM2	34.7	**1.9**	19.6	13.6
MEM3	50.6	19.6	**1.0**	27.8
MEM4	32.1	13.6	27.8	**4.2**

Furthermore, the average distances for the memory components are shown in Table III. The distances are calculated for each iteration of the cross-validation process before being averaged. As the table demonstrates, the class-distances for all four memory components are significantly larger than the self-distances. Therefore, there is little risk of manufacturing variations causing misclassification for the memories.

As a side note, the memory measurements provide an opportunity to examine the impact other factors have on

the measured signal. For example, measurements taken on the two MEM4 components on a A20-LIME2 are shown in Fig. 9. The only difference between the two components is their physical location on the device; however, the spectrums differ noticeably. Visually, it can be difficult to determine whether the spectrums belong to the same family of component. The sidebands generated by the excitation program are much weaker relative to the carrier at U3 compared to U2. Furthermore, the other activity modulating the carrier is stronger at U3. These differences are likely due to differences in PCB traces connected to the components and the relative position of the measurement probe. Despite their differences, both spectrums are correctly identified as being from MEM4. This identification is possible because most of the differences in the two spectrums are minimized after the measurements are projected into the new feature space and the dimensions are reduced to $K = 4$. Since the strongest variation in the data is represented by the first four singular vectors, smaller variations between measurements are lost when decreasing the dimensions.

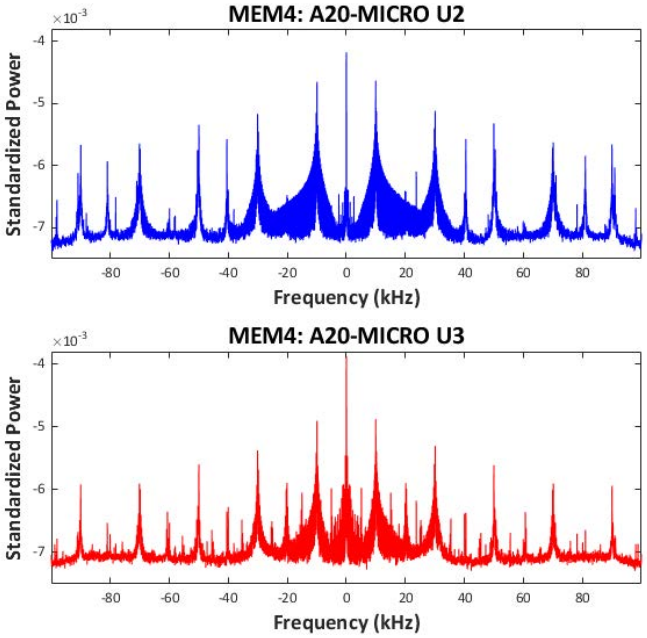


Fig. 9. Comparison of the MEM4 EM signatures from A20-MICRO U2 (top in blue) and A20-MICRO U3 (bottom in red).

E. Recognition of Processor Components

All the IoT devices have their processors integrated into a system-on-chip (SoC) from Allwinner Technology. The specific SoC is identified by the beginning of the device name (Allwinner A10, Allwinner A13, Allwinner A20, and Allwinner A33). Allwinner licensed designs for the processors from ARM Holdings and integrated the design in their SoCs. The A10 and A13 SoCs have a single Cortex-A8 CPU-core [67]. The A20 SoCs have a dual Cortex-A7 CPU-core [68]. The A33 Allwinner has a quad Cortex-A7 CPU-core.

While A10 and A13 have the same type of processor and A20 and A33 have the same type of processor, they still need to be classified as separate components. In both cases, the processors are based on the same design from ARM; however, the processors are not supplied as discrete components directly from ARM. Instead, Allwinner has to implement the design on each SoC. While the functionality may be the same, there will be slight differences in the processor layout based on the other electronics integrated into the SoC and the layout choices of the designer. From the prospective of classification, the differences between how a processor is implemented on difference types of SoC is similar to reversed engineered or tampered components. Therefore, each type of SoC needs to be classified as a unique group. Since the main focus of this section is identifying the processors, the Allwinner A10 is classified as PROC1A, the Allwinner A13 as PROC1B, Allwinner A20 as PROC2A, and Allwinner A33 as PROC2B.

Fig. 10 shows an example of the spectrums measured from an example of each type of processor while they are being excited. The top spectrum is an example of PROC1A from a A10-LIME, the second is an example of PROC1B from an A13-MAIN, the third is example of PROC2A from a A20-LIME1, and the bottom is an example of PROC2B from A33-MAIN. The measurements have been standardized.

All four spectrums have a strong carrier, with harmonics caused by the excitation program at 20 kHz intervals (the even harmonics are too weak to see), giving them the same general shape. Furthermore, the similarities are strongest between SoCs that share the same processor design (i.e. PROC1A and PROC1B are very similar and PROC2A and PROC2B are very similar). However, there are slight differences in properties discussed in Section II-A. For example, the sidebands and carrier for PROC1A are stronger and have a larger spread than the others. This and other differences are magnified after projecting the measurements into the new feature space generated from the training data.

During classification, measurements from A10-LIME, A13-MAIN, A20-LIME1 and A33-MAIN are used as training data for the processors. The overall classification accuracy after cross-validation is 99.5%. A breakdown of the classification results for each device type is shown in Table IV. In the table, the rows correspond to the measured devices and their correct classification, while the columns correspond to the classification determined by the algorithm. The percentage the device is correctly classified appears in bold, while the percentage the device is incorrectly classified appears in red.

As the table demonstrates, the individual classification accuracies for all devices are 98% or higher. Importantly, the algorithm is able to accurately classify each processor regardless of the motherboard it is integrated into. Despite using only examples from A13-MAIN and A20-LIME1 for training, the algorithm correctly classified both A13 devices as having a PROC1B and all three A20 devices as having a PROC2A. At the same time, the algorithm is able to correctly distinguish the A10-LIME from the A13 devices and the A20 devices from the A33-MAIN despite having the similar processors. The differences in how the processor is implemented on the SoC are enough to distinguish them. Furthermore, the algorithm correctly differentiated the processors on the A10-LIME and A20-LIME1 despite the A10-LIME's PCB being an older revision of the A20-LIME1 and the A20 Allwinner being pin-

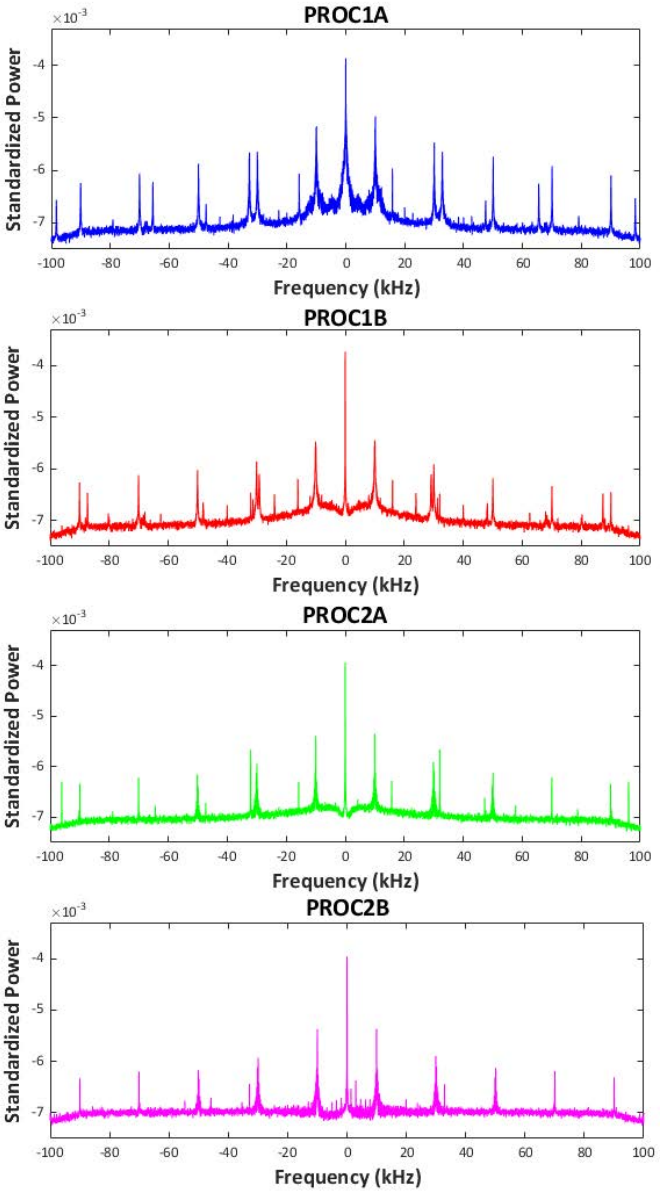


Fig. 10. Comparison of the EM signatures from PROC1A (top in blue), PROC1B (second from the top in red), PROC2A (third from the top in green), and PROC2B (bottom in magenta).

TABLE IV
CONFUSION MATRIX FOR THE PROCESSORS (IN %).

	PROC1A	PROC1B	PROC2A	PROC2B
PROC1A: A10-LIME	**100.0**	0	0	0
PROC1B: A13-MAIN	0	**98.0**	2.0	0
PROC1B: A13-MICRO	0	**100.0**	0	0
PROC2A: A20-LIME1	0	0.2	**99.8**	0
PROC2A: A20-LIME2	0	0	**98.7**	1.3
PROC2A: A20-MICRO	0	0	**100.0**	0
PROC2B: A33-MAIN	0	0	0	**100.0**

to-pin compatible with the A10 Allwinner.

Furthermore, the average distances for the processors are shown in Table V. As the table demonstrates, the class-distances for all four processors are larger than the self-distances, however, not as much as for the memory components.

TABLE V
AVERAGE DISTANCES BETWEEN PROCESSOR COMPONENTS

	PROC1A	PROC1B	PROC2A	PROC2B
PROC1A	**2.7**	12.6	17.3	22.5
PROC1B	12.6	**2.9**	7.5	13.4
PROC2A	17.3	7.5	**3.0**	6.7
PROC2B	22.5	13.4	6.7	**2.4**

F. Recognition of Ethernet Transceiver Components

The Ethernet transceivers consist of three types: ETH1, ETH2, and ETH3. ETH1 is used on A10-LIME, ETH2 is used on A20-LIME1 and A20-MICRO, and ETH3 is used on A20-LIME2. The A13-MAIN, A13-MICRO and A33-MAIN do not have Ethernet transceivers.

Examples of the spectrums for ETH1, ETH2, and ETH3 are shown in Fig. 11. The spectrums are not as active as the memory and processor, indicating that the excitation program is not having as strong of an effect. Furthermore, the signatures for all three components share some similar features. For instance, all three spectrums have activity appearing every 8 kHz with varying magnitudes. However, despite these factors, the differences in the spectrums are still significant enough to distinguish the components. For example, the carrier for ETH1 has a larger spread than the others. At the same time, it has more instances of weak activity distributed throughout the spectrum. On the other hand, the spectrum from ETH2 has more activity within the first 10 kHz of the carrier. Finally, the spectrum for ETH3 has less interference than the other components.

Example measurements from A10-LIME, A20-MICRO, and A20-LIME2 are used for training. The classification accuracies for each individual Ethernet transceiver are shown in Table VI. The overall classification accuracy for the transceivers is 97.7%. As the table demonstrates, all the transceivers had some classification error, the worst being the A20-LIME1 with a total error of 3.8%. The errors are likely the result of several factors. First, the features of the signature are relatively weak, making them more vulnerable to noise. Second, there are variations in signatures from the same type of component, making it difficult for the algorithm to correctly group all the measurements from the same transceivers together. Third, features shared between signatures from different types of transceivers make it difficult for the algorithm to distinguish the different types of transceivers. The classification accuracy could be potentially improved by changing the measurement settings. Some possibilities include increasing the measurement bandwidth (to increase the number of features for classification), increasing the measurement time (to decrease the influence of noise), or using a different set of instructions for exciting the component.

The average distances for the Ethernet transceivers are shown in Table VII. As the table demonstrates, the class-distances are larger than the self-distances for each transceiver.

VI. CONCLUSIONS

This work proposes leveraging EM side-channels to recognize/authenticate components integrated onto a motherboard.

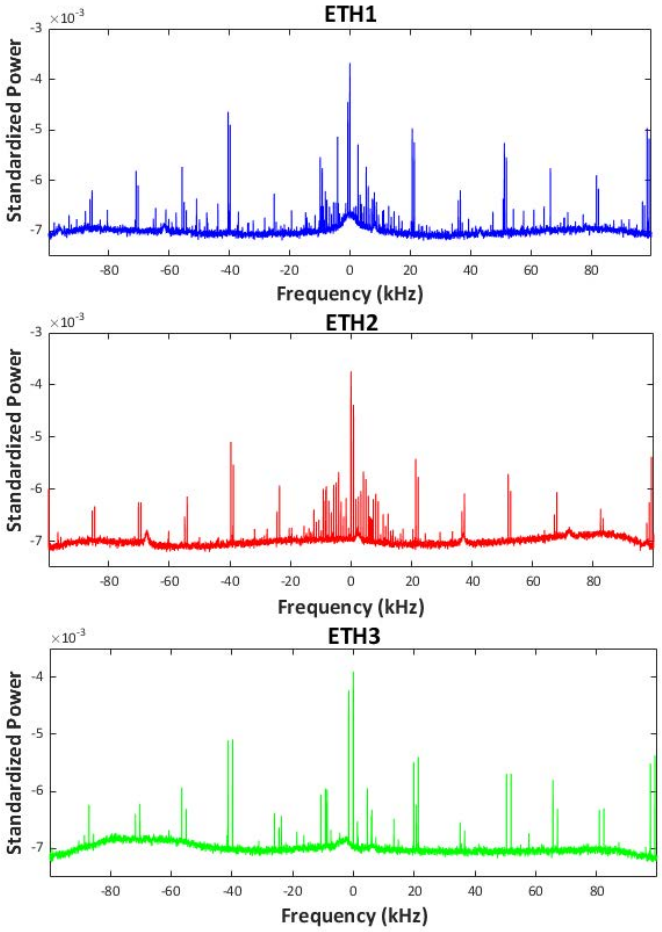


Fig. 11. Comparison of the EM signatures from ETH1 (top in blue), ETH2 (middle in red), and ETH3 (bottom in green).

TABLE VI
CONFUSION MATRIX FOR ETHERNET TRANSCEIVERS (IN %)

	ETH1	ETH2	ETH3
ETH1: A10-LIME	**98.2**	1.8	0
ETH2: A20-LIME1	2.3	**96.2**	1.5
ETH3: A20-LIME2	0	3.2	**96.8**
ETH2: A20-MICRO	0	**99.7**	0.3

By focusing on components on a motherboard, our method provides an opportunity to authenticate devices assembled by third parties. The proposed method identifies components based on the modulated signals emanated during the component's operation. These signals are generated by exciting the component in a controlled and repeatable manner. When testing an unknown component, the spectrum is compared to previously recorded signatures taken during training. To improve the efficiency of the proposed method, the size of the spectrum is first reduced by projecting it into a vector space generated from training signatures. The identity of the tested component is then determined using a k-NN algorithm. The proposed method has successfully classified components such as external memories, processors, and Ethernet transceivers integrated on seven types of IoT devices. Nine to ten different instances of each device are used in the experiments and then cross-validated during classification. Cross-type testing of the motherboards is conducted as well. Since manufacturers commonly use the same components in multiple designs, being able to collect the training signatures on one motherboard and test components from different motherboards significantly speeds up the process and decreases the cost. Using the measurements taken while the components were excited for 1 s, our method achieved a classification accuracy greater than 96% for all the tested components. These results demonstrate that this method can recognize components based on their EM emanations even if they are integrated on a completely different motherboard.

TABLE VII
AVERAGE DISTANCES BETWEEN ETHERNET TRANSCEIVERS

	ETH1	ETH2	ETH3
ETH1	**3.5**	11.5	14.4
ETH2	11.5	**3.5**	5.4
ETH3	14.4	5.4	**2.5**

REFERENCES

[1] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, Aug 2014.

[2] N. Kae-Nune and S. Pesseguier, "Qualification and testing process to implement anti-counterfeiting technologies into IC packages," in *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2013, pp. 1131–1136.

[3] M. Pecht and S. Tiku, "Bogus: electronic manufacturing and consumers confront a rising tide of counterfeit electronics," *IEEE spectrum*, vol. 43, no. 5, pp. 37–46, 2006.

[4] H. Dogan, D. Forte, and M. M. Tehranipoor, "Aging analysis for recycled FPGA detection," in *2014 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, Oct 2014, pp. 171–176.

[5] C. Jin and M. van Dijk, "Secure and efficient initialization and authentication protocols for SHIELD," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 1, pp. 156–173, 2019.

[6] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The EM side—channel (s)," in *International workshop on cryptographic hardware and embedded systems*. Springer, 2002, pp. 29–45.

[7] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *International workshop on cryptographic hardware and embedded systems*. Springer, 2001, pp. 251–261.

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

[9] M. Backes, M. Dürmuth, S. Gerling, M. Pinkal, and C. Sporleder, "Acoustic side-channel attacks on printers." in *USENIX Security symposium*, 2010, pp. 307–322.

[10] M. Hutter and J.-M. Schmidt, "The temperature side channel and heating fault attacks," in *International Conference on Smart Card Research and Advanced Applications*. Springer, 2013, pp. 219–235.

[11] L. N. Nguyen, C.-L. Cheng, M. Prvulovic, and A. Zajić, "Creating a backscattering side channel to enable detection of dormant hardware trojans," *IEEE transactions on very large scale integration (VLSI) systems*, vol. 27, no. 7, pp. 1561–1574, 2019.

[12] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, RSA, DSS, and other systems," in *Annual International Cryptology Conference*. Springer, 1996, pp. 104–113.

[13] M. Alam, H. A. Khan, M. Dey, N. Sinha, R. Callan, A. Zajić, and M. Prvulovic, "One&done: A single-decryption EM-based attack on OpenSSL's constant-time blinded RSA," in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 585–602.

[14] W. Van Eck, "Electromagnetic radiation from video display units: An eavesdropping risk?" *Computers & Security*, vol. 4, no. 4, pp. 269–286, 1985.

[15] M. G. Kuhn, "Compromising emanations: eavesdropping risks of computer displays," Ph.D. dissertation, University of Cambridge, 2002.

[16] D. Genkin, I. Pipman, and E. Tromer, "Get your hands off my laptop: Physical side-channel key-extraction attacks on pcs," *Journal of Cryptographic Engineering*, vol. 5, no. 2, pp. 95–112, 2015.

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

[18] N. Sehatbakhsh, A. Nazari, M. Alam, F. Werner, Y. Zhu, A. Zajić, and M. Prvulovic, "REMOTE: Robust external malware detection framework by using electromagnetic signals," *IEEE Transactions on Computers*, 2019.

[19] H. A. Khan, N. Sehatbakhsh, L. N. Nguyen, M. Prvulovic, and A. Zajić, "Malware detection in embedded systems using neural network model for electromagnetic side-channel signals," *Journal of Hardware and Systems Security*, vol. 3, no. 4, pp. 305–318, 2019.

[20] L. N. Nguyen, C.-L. Cheng, F. T. Werner, M. Prvulovic, and A. Zajić, "A comparison of backscattering, EM, and power side-channels and their performance in detecting software and hardware intrusions," *Journal of Hardware and Systems Security*, pp. 1–16, 2020.

[21] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using IC fingerprinting," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, 2007, pp. 296–310.

[22] M. M. Ahmed, D. Hely, N. Barbot, R. Siragusa, E. Perret, M. Bernier, and F. Garet, "Towards a robust and efficient EM based authentication of FPGA against counterfeiting and recycling," in *2017 19th International Symposium on Computer Architecture and Digital Systems (CADS)*, Dec 2017, pp. 1–6.

[23] 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 *2020 SPIE Defense + Commercial Sensing Digital Forum*, 2020, to be published.

[24] X. Dong, H. Weng, D. G. Beetner, T. H. Hubing, D. C. Wunsch, M. Noll, H. Göksu, B. Moss *et al.*, "Detection and identification of vehicles based on their unintended electromagnetic emissions," *IEEE Transactions on Electromagnetic Compatibility*, vol. 48, no. 4, pp. 752–759, 2006.

[25] H. Göksu, D. C. Wunsch, X. Dong, A. Kökce, and D. G. Beetner, "Detection and identification of vehicles based on their spark-free unintended electromagnetic emissions," *IEEE Transactions on Electromagnetic Compatibility*, vol. 60, no. 5, pp. 1594–1597, 2018.

[26] C. Herder, M. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, Aug 2014.

[27] J. M. Vann, T. P. Karnowski, R. Kerekes, C. D. Cooke, and A. L. Anderson, "A dimensionally aligned signal projection for classification of unintended radiated emissions," *IEEE Transactions on Electromagnetic Compatibility*, vol. 60, no. 1, pp. 122–131, 2017.

[28] J. M. Vann, T. Karnowski, and A. L. Anderson, "Classification of unintended radiated emissions in a multi-device environment," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5506–5513, 2018.

[29] C. Yang and A. P. Sample, "EM-ID: Tag-less identification of electrical devices via electromagnetic emissions," in *2016 IEEE International Conference on RFID (RFID)*. IEEE, 2016, pp. 1–8.

[30] G. Laput, C. Yang, R. Xiao, A. Sample, and C. Harrison, "EM-Sense: Touch recognition of uninstrumented, electrical and electromechanical objects," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, 2015, pp. 157–166.

[31] M. M. Ahmed, D. Hely, E. Perret, N. Barbot, R. Siragusa, M. Bernier, and F. Garet, "Authentication of microcontroller board using non-invasive EM emission technique," in *2018 IEEE 3rd International Verification and Security Workshop (IVSW)*. IEEE, 2018, pp. 25–30.

[32] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[33] R. Callan, A. Zajić, and M. Prvulovic, "FASE: finding amplitude-modulated side-channel emanations," in *2015 ACM IEEE 42nd Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2015, pp. 592–603.

[34] M. Prvulovic, A. Zajić, R. L. Callan, and C. J. Wang, "A method for finding frequency-modulated and amplitude-modulated electromagnetic emanations in computer systems," *IEEE Transactions on Electromagnetic Compatibility*, vol. 59, no. 1, pp. 34–42, 2016.

[35] R. Callan, A. Zajić, and M. Prvulovic, "A practical methodology for measuring the side-channel signal available to the attacker for instruction-level events," in *2014 47th Annual IEEE ACM International Symposium on Microarchitecture*. IEEE, 2014, pp. 242–254.

[36] F. Werner, D. A. Chu, A. R. Djordjević, D. I. Olćan, M. Prvulovic, and A. Zajić, "A method for efficient localization of magnetic field sources excited by execution of instructions in a processor," *IEEE Transactions on Electromagnetic Compatibility*, vol. 60, no. 3, pp. 613–622, 2017.

[37] C. Wang, R. Callan, A. Zajić, and M. Prvulovic, "An algorithm for finding carriers of amplitude-modulated electromagnetic emanations in computer systems," in *2016 10th European Conference on Antennas and Propagation (EuCAP)*. IEEE, 2016, pp. 1–5.

[38] R. Callan, N. Popovic, A. Daruna, E. Pollmann, A. Zajić, and M. Prvulovic, "Comparison of electromagnetic side-channel energy available to the attacker from different computer systems," in *2015 IEEE International Symposium on Electromagnetic Compatibility (EMC)*. IEEE, 2015, pp. 219–223.

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

[40] EM probe station. Riscure. [Online]. Available: https:\\getquote.riscure.com\en\quote\2101064\em-probe-station.htm

[41] C. Beleites, R. Baumgartner, C. Bowman, R. Somorjai, G. Steiner, R. Salzer, and M. G. Sowa, "Variance reduction in estimating classification error using sparse datasets," *Chemometrics and intelligent laboratory systems*, vol. 79, no. 1-2, pp. 91–100, 2005.

[42] M. Dey, A. Nazari, A. Zajić, and M. Prvulovic, "EMPROF: Memory profiling via EM-emanation in IoT and hand-held devices," in *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2018, pp. 881–893.

[43] A10-olinuxino-lime. Olimex. [Online]. Available: https://www.olimex.com/Products/OLinuXino/A10/A10-OLinuXino-LIME-n4GB/open-source-hardware

[44] A13-olinuxino. Olimex. [Online]. Available: https://www.olimex.com/Products/OLinuXino/A13/A13-OLinuXino/open-source-hardware

[45] A13-olinuxino-micro. Olimex. [Online]. Available: https://www.olimex.com/Products/OLinuXino/A13/A13-OLinuXino-MICRO/open-source-hardware

[46] A20-olinuxino-lime. Olimex. [Online]. Available: https://www.olimex.com/Products/OLinuXino/A20/A20-OLinuXino-LIME/open-source-hardware

[47] A20-olinuxino-lime2. Olimex. [Online]. Available: https://www.olimex.com/Products/OLinuXino/A20/A20-OLinuXino-LIME2/open-source-hardware

[48] A20-olinuxino-micro. Olimex. [Online]. Available: https://www.olimex.com/Products/OLinuXino/A20/A20-OLinuXino-MICRO/open-source-hardware

[49] A33-olinuxino. Olimex. [Online]. Available: https://www.olimex.com/Products/OLinuXino/A33/A33-OLinuXino/open-source-hardware

[50] Olinuxino is open source. Olimex. [Online]. Available: https://github.com/OLIMEX/OLINUXINO

[51] DE0-CV cyclone V board. Zentel Electronics Corp. [Online]. Available: https://www.intel.com/content/www/us/en/programmable/solutions/partners/partner-profile/terasic-inc-/board/de0-cv-cyclone-v-board.html

[52] Altera DE1 board. Intel Corporation. [Online]. Available: https://www.intel.com/content/www/us/en/programmable/solutions/partners/partner-profile/terasic-inc-/board/altera-de1-board.html

[53] K4B4G1646E-BYK0. Samsung. [Online]. Available: https://www.samsung.com/semiconductor/dram/ddr3/K4B4G1646E-BYK0/

[54] H5TQ2G83FFR. Hynix Semiconductor. [Online]. Available: https://www.skhynix.com/eolproducts.view.do?pronm=DDR3+SDRAM&srnm=H5TQ2G83FFR&rk=19&rc=consumer

[55] H5TQ2G63BFR. Hynix Semiconductor. [Online]. Available: https://www.skhynix.com/eolproducts.view.do?pronm=DDR3+SDRAM&srnm=H5TQ2G63BFR&rk=19&rc=computing

[56] MT41K256M16HA-125 XIT. Micron. [Online]. Available: https://www.micron.com/products/dram/ddr3-sdram/part-catalog/mt41k256m16ha-125-xit

[57] IS42/45R86400D/16320D/32160D IS42/45S86400D/16320D/32160D. Integrated Silicon Solution Inc. [Online]. Available: http://www.issi.com/WW/pdf/42-45R-S_86400D-16320D-32160D.pdf

[58] A3V64S40GTP/GBF. Zentel Electronics Corporation. [Online]. Available: https://zentel-europe.com/datasheets/A3V64S40GTP_v1.3_Zentel.pdf

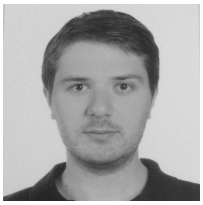
[59] Allwinner A10. Allwinner Technology. [Online]. Available: https://web.archive.org/web/20160729114202/http://www.allwinnertech.com/en/clq/processora/A10.html

[60] Allwinner A13. Allwinner Technology. [Online]. Available: https://web.archive.org/web/20160811122523/http://www.allwinnertech.com/en/clq/processora/A13.html

[61] Allwinner A20. Allwinner Technology. [Online]. Available: http://www.allwinnertech.com/index.php?c=product&a=index&id=45

[62] Allwinner A33. Allwinner Technology. [Online]. Available: http://www.allwinnertech.com/index.php?c=product&a=index&id=23
[63] RTL8201CP. Realtek. [Online]. Available: http://realtek.info/pdf/rtl8201cp.pdf
[64] LAN8710A. Microchip Technology. [Online]. Available: https://www.microchip.com/wwwproducts/en/LAN8710A
[65] KSZ9031. Microchip Technology. [Online]. Available: https://www.microchip.com/wwwproducts/en/KSZ9031
[66] J. H. Anderson, "A PUF design for secure FPGA-based embedded systems," in *2010 15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2010, pp. 1–6.
[67] Cortex-a8. ARM Developer. [Online]. Available: https://developer.arm.com/ip-products/processors/cortex-a/cortex-a8
[68] Cortex-a7. ARM Developer. [Online]. Available: https://developer.arm.com/ip-products/processors/cortex-a/cortex-a7

Frank T. Werner received his B.S. degree (2013) and his M.S. (2016) in electrical engineering from Auburn University, Alabama. Currently, he is completing his PhD in electrical engineering at Georgia Institute of Technology. His research interests include electromagnetic compatibility, wireless communications, signal processing, and applied electromagnetics.



Baki Berkay Yilmaz (S'16) received the B.Sc. and M.Sc. degrees in Electrical and Electronics Engineering from Koc University, Turkey in 2013 and 2015 respectively. He joined Georgia Institute of Technology in fall 2016, and he is currently pursuing his PhD in the School of Electrical and Computer Engineering, focusing on quantifying covert/side-channel information leakage. Previously, he worked on channel equalization and sparse reconstruction. His research interests span areas of electromagnetics, signal processing, and information theory.

Milos Prvulovic (S'97-M'03-SM'09) received the B.Sc. degree in electrical engineering from the University of Belgrade in 1998, and the M.Sc. and Ph.D. degrees in computer science from the University of Illinois at Urbana-Champaign in 2001 and 2003, respectively. He is an Associate Professor in the School of Computer Science at the Georgia Institute of Technology, where he joined in 2003. His research interests are in computer architecture, especially hardware support for software monitoring, debugging, and security.

Dr. Prvulovic is recipient of the following awards/honors: NSF CAREER Award (2005), Best Paper Award at the 49th Annual IEEE/ACM International Symposium on Microarchitecture, 2016, Distinguished Alumni Educator Award, 2012, from the Department of Computer Science at the University of Illinois at Urbana-Champaign, Hesburgh Teaching Fellowship, 2010-2011, Intel PhD Fellowship, 2002, W.J. Poppelbaum Memorial Award, 2001.

Alenka Zajić (S'99-M'09-SM'13) received the B.Sc. and M.Sc. degrees form the School of Electrical Engineering, University of Belgrade, in 2001 and 2003, respectively. She received her Ph.D. degree in Electrical and Computer Engineering from the Georgia Institute of Technology in 2008. Currently, she is an Associate Professor in the School of Electrical and Computer Engineering at Georgia Institute of Technology. Her research interests span areas of electromagnetics, wireless communications, signal processing, and computer engineering.

Dr. Zajić is the recipient of the following awards: NSF CAREER Award (2017), Best Paper Award at the 49th Annual IEEE/ACM International Symposium on Microarchitecture, 2016, 2012 Neal Shepherd Memorial Best Propagation Paper Award, the Best Student Paper Award at the IEEE International Conference on Communications and Electronics 2014, the Best Paper Award at the International Conference on Telecommunications 2008, the Best Student Paper Award at the 2007 Wireless Communications and Networking Conference, LexisNexis Dean's Excellence Award 2016, Richard M. Bass/Eta Kappa Nu Outstanding Teacher Award 2016. She has been an editor for IEEE Transactions on Wireless Communications 2012-2017.