DYNAMIC POWER CONSUMPTION MONITORING IN SDR AND CR REGULATORY COMPLIANCE

Carlos R. Aguayo Gonzalez (MPRG, Wireless@Virginia Tech, Blacksburg, VA, USA; caguayog@vt.edu); and Jeffrey H. Reed (MPRG, Wireless@Virginia Tech, Blacksburg, VA, USA; reedjh@vt.edu).

ABSTRACT

Software-defined radios (SDRs) present a new paradigm in the implementation and resource management of wireless communication systems. In doing so, they introduce increased interference risks because of both their ability to access wide spectral bands and their vulnerability to software configuration errors and malicious attacks. A necessary element for SDR mainstream deployment, is the formulation of adequate policies to prevent unauthorized software changes without placing unreasonable regulatory burdens on designers and manufacturers

In this paper, we introduce novel approach for assessing the integrity of SDR software execution by monitoring their dynamic power consumption. The approach relies on extracting distinctive power consumption features and then determining whether they correspond to authorized behavior by using pattern recognition techniques. This approach provides a mechanism to detect the execution of unauthorized software and support regulatory compliance.

Preliminary results show the correct identification of basic software routines executing on different platforms. Discriminatory features are extracted in the time domain and from the execution on basic evaluation boards. These results corroborate the existence of power fingerprints and motivate further research on this topic.

1. INTRODUCTION

Software-defined radio (SDR) and cognitive radio (CR) have revolutionized the way we manage radio resources and have enabled myriad new applications due to its flexibility and upgradeability. In order to achieve the full potential of these technologies, it is necessary that other critical aspects also keep pace with SDR and CR developments—in particular the regulatory aspects. Regulators are currently faced with the extraordinary challenge of effectively manage the increased interference risks brought by SDR and CR without hindering innovation in communications technology. It is necessary to develop the mechanisms that facilitate manufacturers demonstrate regulatory compliance and allow regulators to enforce policies.

There are some fundamental differences between traditional radio systems and SDR and CR that impact the way policies are defined and enforced. For a legacy radio system, with a small number of operational modes limited by hardware, it is sufficient to define limits on spectral emissions and certify compliance at development time. In case of misuse or malfunctioning, the impact is limited to the specific spectral bands defined by hardware. The interference risks are higher with SDR because they contain flexible hardware that can access wide spectral bands and support several different behaviors and modes. CR will raise the stakes even higher. With these devices, a simple software modification can completely change the operational properties of the radio, multiplying the certification requirements and greatly increasing the risks associated with the technology.

As a result of the increase in interference risks, regulatory bodies have been cautious in developing policy and authorizing SDR and CR. The Federal Communications Commission (FCC) has developed policy that allows streamlined authorization for software changes that affect spectral emissions in SDR, but it still requires thorough compliance testing. As part of the SDR certification process, manufacturers are required to demonstrate that only authorized software is allowed to execute. No particular mechanism has been specified by the FCC to achieve this. For CR, there have been several discussions to define adequate policy but no agreement has been reached yet.

In this paper, we present a novel integrity assessment approach that can leverage SDR and CR regulatory policy compliance and enforcement. Our proposed approach has the potential to deliver a mechanism for run-time monitoring of SDR and CR regulatory compliance. It relies on extracting power consumption signatures, or fingerprints, from the execution of authorized software and then using pattern recognition techniques to determine the continuous compliance at run time. When applied successfully, this approach allows an external monitor to detect the execution of precharacterized critical modules and, under the right circumstances, even certain execution parameters can be identified.

2. SDR AND CR REGULATORY FRAMEWORK

Under the current regulatory framework, traditional radios are not allowed to modify their operating frequency, output power, or types of radio frequency emissions without going through a re-certification process. This is because such changes involve hardware modifications that practically yield a new device. With SDR and CR, these changes can be made with just a software upgrade. Therefore, regulatory bodies and manufacturers were forced to devise new policies, certification techniques, and enforcement mechanisms in an attempt to maintain the delicate balance between interference management and regulatory burden. This has been difficult because of the the complexity and early state of SDR and CR technologies, which has not allowed regulatory bodies time to determine the adequacy of adopted policies.

Currently, the US FCC has adopted new policies to regulate SDR [1]. In the new rules, the FCC amended its

> *equipment authorization rules to permit equipment manufacturers to make changes in the frequency, power, and modulation parameters of such radios (SDR) without the need to file a new equipment authorization with the Commission. We (the FCC) will also permit electronic labeling so that a third party may modify a radio's technical parameters without having to return it to the manufacturer for relabeling*

This policy designates a new class of permissive changes. "*Any changes in frequency, power, or modulation type of a software defined radio may be authorized as a Class III permissive change*". This designation streamlines the filing procedure for changes to approved SDR and eliminates the need for new identification numbers. The manufacturer, however, still needs to submit test data showing that the equipment complies with the applicable rule parts and RF exposure requirements with the new software.

This policy, however, limits certification to specific hardware-software pairs to prevent unknown effects on the RF emissions of a radio. The only hardware changes allowed with a Class III permissive change are those in which the hardware modifications do not affect radio frequency emissions. This policy also requires manufacturers to prevent unauthorized software changes that could affect the compliance of a radio. This was never a concern with legacy radios. The FCC stopped short of mandating specific requirements for this authentication. Instead, it is left to the manufacturers to "*take steps to ensure that only software that is part of hardware/software combination approved by the Commission can be loaded into a radio*". The Commission leaves open the possibility of specifying more detailed security requirements at a later date as SDR technology improves

This policy seems to be adequate for current systems. Unfortunately, there is no way to monitor the continuous compliance of devices over time. As SDR technology becomes more prevalent and becomes target of more malicious attacks, it will be necessary to continuously check for software integrity. Furthermore, the complexity of regulating future radio systems is becoming more and more evident with the recent development of CR technology. CR are based on flexibility and adaptability to environmental or usage changes. Because frequency agility is a key technical feature of CR, the command-and-control approach of the current regulatory framework may drastically limit the potential benefits of the technology. Allowing frequency agile devices to operate under the current regulatory framework would require the identification during the certification process of the devices that could potentially abuse its privileges, either intentionally or by accident. We are likely to come out with our hands empty trying to solve this question.

Several approaches have been proposed to allow devices with dynamic spectrum access to operate while still being able to enforce regulatory policies at a reasonable cost. Sahai et al. [2] have proposed an approach for light-handed regulation of CR. The proposed approach gives CR relative flexibility in terms of spectral emissions provided they embed in their transmissions a unique signature that allows them to be identified and punished in the event of misbehavior. This signature is expected to be easy to certify and implement, and which does not assume a specific waveform to be implemented by the radio. In the proposed approach, this identity is implemented by "giving each radio its own spectral fingerprint of time-frequency slots that it is forbidden to use". This type of signature allows harmful interference to be attributed to the misbehaving radio, which is in turn banned from using the spectrum for a finite amount of time. The argument in favor of this approach is that it is easy to certify since it only requires a trusted device that disables transmission.

Another approach has been proposed by Chapin and Lehr [3]. In this approach, time-limited leases (TLL) grant a set of transmission right to the radios that hold them. TLL are particularly useful in cases where the right holders are difficult to locate, as it would be the case in CR networks. The implementation of TLL would rely on "manufacturers including in their devices a simple, secure subsystem that contains a clock and controls critical features such as transmitter power and frequency settings. The subsystem has enough computing power to validate cryptographically-signed lease extension messages. It disables specified radio features if no extension message has been received by the end of the lease period".

Both approaches can be complimentary. Furthermore, they are both somewhat related in the sense that they rely on a trusted module in charge of enforcing the regulatory compliance. It is this module which needs to be certified and, when possible, monitored in future radios.

3. PROPOSED APPROACH

Dynamic power consumption (DPC) in a digital circuit is due to transient currents during switching and charge and discharge of load capacitances [4]. In a digital processor, DPC depends on the instructions being executed, their parameters and addresses, as well as inter-instruction transitions. As a result, the execution of a given software routine would yield a specific DPC pattern or fingerprint.

The envisioned monitor, depicted in Figure 1, is placed between the power supply and the main processing hardware. This monitor can provide a mechanism to enforce that only certified software and/or hardware modules are executed. During device certification, the power fingerprint of the certified software is stored independently and then used at runtime to verify that only the software used during certification is executed. Regulatory entities can leverage their current mechanisms with this approach to ensure that critical modules that impact spectral emissions are not modified after deployment.

This approach can be used to prevent the execution of SDR with unauthorized software, as required by the FCC [1] and for which there is currently no standard procedure. For

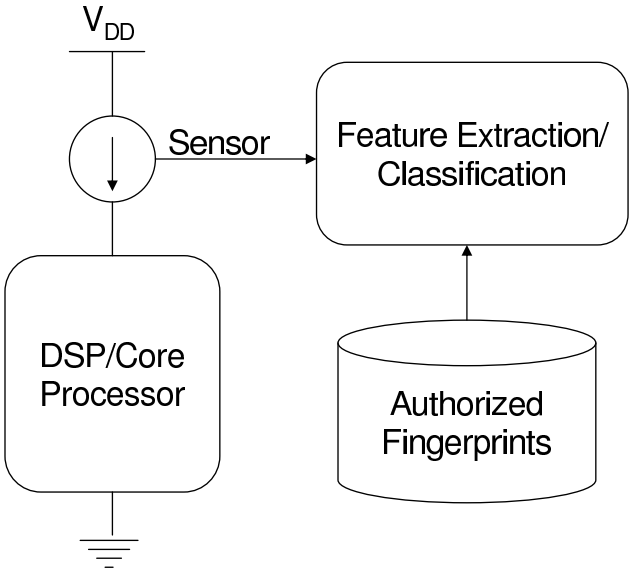

Fig. 1. Proposed system description

CR, this approach can be useful for enforcing the execution of critical modules, such as those controlling the time-limited leases or the embedded spectral fingerprints, giving regulators a run-time enforcement mechanism for dynamic radio communication systems.

To implement this approach, it is necessary to perform three basic tasks common to all pattern recognition systems: sensing, preprocessing and feature extraction, and classification and decision making.

3.1. Sensing Mechanisms

Sensing, in this case, involves measuring, directly or indirectly, the instantaneous current being drained by the digital hardware. For example, it can be performed with a commercial current probe and a digital oscilloscope. It can also be performed with a shunt resistor in series with the core power supply or with a modified Wilson current mirror, as explained in [5] [6]. In both cases, the voltage differential across measuring resistors can be captured with an oscilloscope.

The closer the sensing mechanism is placed to the processor core, the less interference from other board elements is allowed. Furthermore, the sensing process needs to be performed in an efficient way if it is to be deployed in embedded devices. Both approaches, shunt resistor and current mirror, can potentially to be implemented as a built-in mechanism in fielded devices, even for small form factors.

There are two important aspects that need to be considered: sensing requirements and sensor location. Sensing requirements depend the specific discriminatory features selected and the current monitor has to be physically located within the target platform. Once sampled, the traces can be transmitted to a different location or stored for posterior processing. While the actual feature extraction and classification algorithms can be performed remotely, the target platform at least must include provisions (contact points or pins) for a sensor to be physically connected.

3.2. Feature Extraction

This approach is based on extracting discriminatory features from the processor's power consumption during execution of authorized software. These are used as a reference to design optimal classifiers and detectors. Similar to other pattern identification systems, the performance of this approach depends on the discriminatory qualities of the features selected. Discriminatory qualities depend on the specific characteristics of the hardware platforms and the software itself.

For simple systems, it may be possible to characterize all allowed execution patterns. For more complex systems, however, it may be necessary to take into consideration multiple features to determine if the software executing is allowed. This may include features obtained from observing multiple complete execution periods and features from observing individual sub-modules. It is important to note that, due to the statistical nature of different discriminatory features, several executions cycles may need to be observed in order to provide adequate performance.

Discriminatory qualities can be improved by considering analysis in different domains, which can yield different resolutions, as well as different sensing and processing requirements. For example, in the time domain, a simple approach would be to align and average power traces captured from several executions of the target software. In this case, a simple correlation operation would yield a decision metric. There are several more options for feature extraction in the frequency and cyclostationary domains. Furthermore, powerful analysis techniques such as principal component analysis and wavelet analysis can be applied to further enhance the performance.

3.3. Categorization and Decision Making

Several algorithms and decision function structures developed for pattern recognition that can be used to determine whether a power trace corresponds to the execution of authorized code. For example, we can use a Bayes classifiers, the K-means algorithm or a neural network. Precharacterization, or training, traces from the execution of authorized software are used to determine pattern classes and decision functions in a supervised way.

4. Preliminary Results

This section presents the results of two feasibility experiments for the proposed approach. For the first experiment, the goal is to detect the execution of precharacterized code on a basic platform using simple cross-correlation in the time domain. An extended version of these results appears in [7]. The second experiment extends the first one by detecting the execution of target code in a processor using fingerprints obtained from a different processor of the same family.

4.1. Time-Domain Correlation Analysis

For this experiment, the target platform is a FOX11 Trainer board [8] which contains a Motorola's HC11 processor (68HC11E1), running at 2 MHz, with 32 KB RAM, 32 KB EPROM, and 32 KB EEPROM. The power traces are taken by measuring the voltage differential across a 1 Ω resistor in series with the main power supply with a Tektronix' TDS 694C digital real-time oscilloscope. The oscilloscope is configured to 500 MS/s, 10 mV per division with a vertical offset of 229 mV and a capture length of 30000 points. The captured trace is transfered to the host computer using GPIB and all the post-processing is executed using MATLAB.

The target code are Lines 12-17 in Listing 1. During the execution of this code, all the bits in Accumulator A are toggled from 0x00 to 0xFF and back several times using different instructions. This code is expected to have a strong power signature given the relatively large number of bits switching at once.

Listing 1. Precharacterization Code

```
1     clra                ;a = 00
2     back: ldab    #$ff    ;LEDs ON
3           stab    portb
4           nop
5           nop
6           nop
7           nop
8           nop
9           ldab    #$00    ;LEDs OFF
10          stab    portb
11          nop
12          eora    #$ff    ;a = ff
13          ldaa    #$00    ;a = 00
14          oraa    #$ff    ;a = ff
15          eora    #$ff    ;a = 00
16          ldaa    #$ff    ;a = ff
17          clra            ;a = 00$
18          nop
19          nop
20           ;.
21           ;. x 15
22           ;.
23          nop
24          nop
25          jmp     back
```

Before the target code, in lines 2-10, there is a "marker" where the board's LEDs are turned on and off. This creates a current drain offset used to trigger the oscilloscope and provides timing information for feature extraction. After the target code, there is a sequence of "nops"[1] to create a delay before repeating the loop.

A trace captured from a single run of this code provides the fingerprint used to determine the execution of the target code. The captured power trace is shown in Figure 2, where the current offset due to the marker indicates the beginning of the loop. Using cycle information from the processor's documentation, the segment of the trace corresponding to the execution of the target code is determined (the markers at the bottom of the plot indicate the estimated processor clock cycles). After filtering out the low frequency components[2] with a high-pass filter with cutoff frequency of 1.5 MHz, the specific segment used as the power fingerprint is extracted.

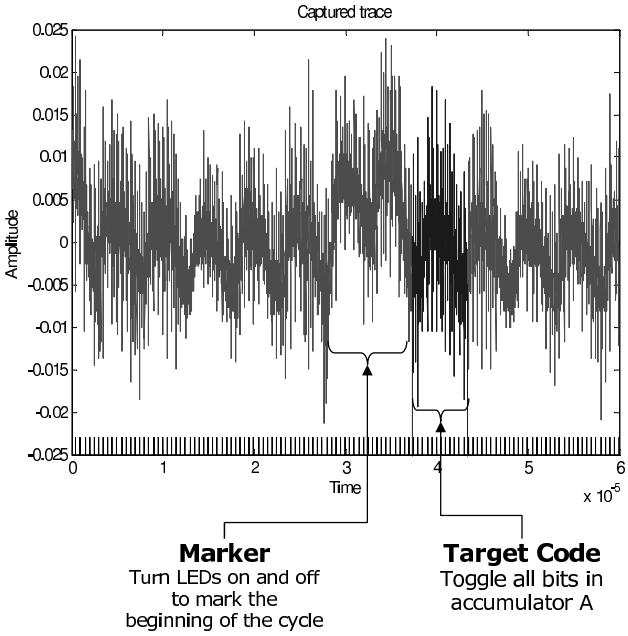

Fig. 2. Captured Precharacterization Power Trace

By cross-correlating this power signature

$$S = \{s_0, s_1, \ldots, s_{L-1}\}$$

against traces captured from different runs of the same code

$$R = \{r_0, r_1, \ldots, r_{K-1}\};\ K \geq L$$

the signature can be easily identified. The cross correlation for different sample lags, k, of the traces is given by:

$$\rho_{SR}(k) = \frac{\sum_{n=0}^{L-1} s_n r_{k+n} - L\bar{S}\bar{R}}{(L-1)\sigma_S\sigma_R}; \qquad 0 \leq k \leq K-L$$

where $\bar{S}$ and σ_S are the sample mean and standard deviation of S, respectively, and $\bar{R}$ and σ_R are the sample mean and standard deviation of the subsequence $\{r_{k+0}, r_{k+1}, \ldots, r_{k+L-1}\} \in R$.

Figure 3(a) shows the correlation of the original power fingerprint with the same trace from where it was extracted. Hence, the perfect correlation. Figure 3 b) shows the cross-correlation with subsequent runs of the same code. The correlation spikes appearing at the expected loop repetition rate indicate the execution of the target code.

In order to confirm correct identification, the instructions around the target code are modified. The number of "nops" inside the marker and after the target code is reduced, yielding a shorter period between loop iterations. Note that this change increases the frequency at which the target code is invoked without affecting the target code itself. The correlation results with this modified code can be seen in Figure 4. As expected, the correlation peaks happen more often, consistent with the shorter duration of the loop.

[1]The NOP instruction means "no operation" and no registers, other than the PC, are affected by it.

[2]The origin of these low frequency components is not completely known. It is believed that they are due to the board's voltage regulation circuitry.

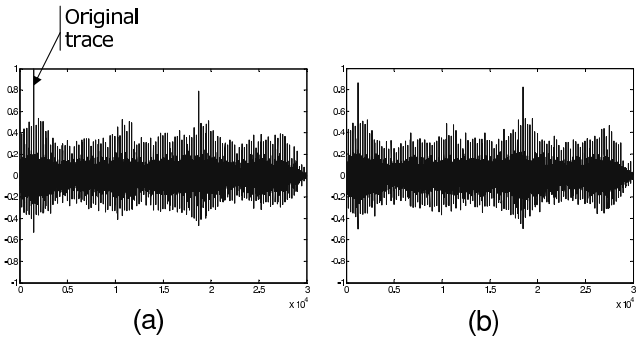

Fig. 3. Cross-correlation results with traces from the original (a) and subsequent runs (b)

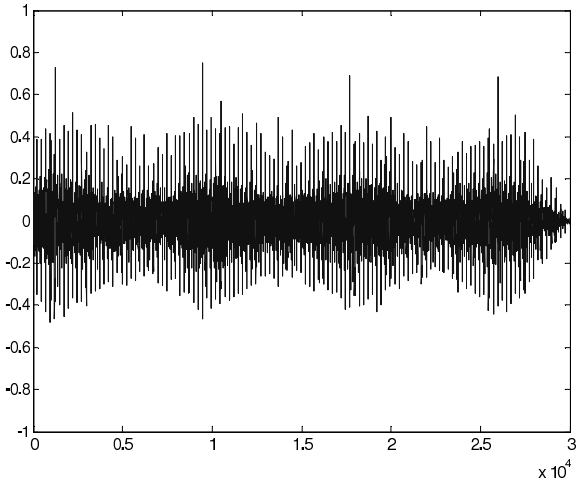

Fig. 4. Cross-Correlation with modified loop (short delay and marker)

One critical aspect that we need to demonstrate is the ability to determine when the target code is not executed. To test this, a modified version of the target code is developed with the same functionality (i.e. the contents of Accumulator A are toggled in the same order and the same number of times) but using a different set of instructions, as shown in Listing 2. These modifications maintain most of the intrinsic periodicities of the code (the frequency of the loop and the duration of the target code) as well as the logical behavior in an effort to make its power fingerprint mimic that of the original code, increasing the risk of false identification by our approach.

Cross-correlating the original signature with traces captured from the execution of the modified code we notice a lack strong peaks, as shown in Figure 5(a). This indicates that the original code did not execute. When correlating with a power signature obtained from the new set of instructions, however, the strong peaks show up again, as seen in Figure 5(b), corroborating the correct identification of target code execution.

Listing 2. Modified Loop Instructions

```
12          oraa    #$ff    ;a = ff
13          clra            ;a = 00
14          ldaa    #$ff    ;a = ff
15          anda    #$00    ;a = 00
16          eora    #$ff    ;a = ff
17          ldaa    #$00    ;a = 00$
```

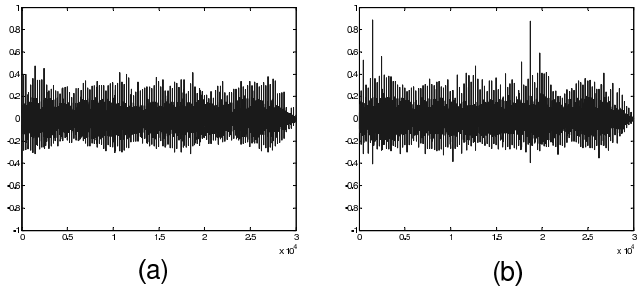

Fig. 5. a)Cross-Correlation with different instructions in loop. b)Cross-correlation with a signature obtained from the modified code

4.2. Cross-Processor Detection

The objective of this experiment is to provide anecdotal evidence of the ability of this approach to detect the execution of target code in a processor using a fingerprint extracted from a different one. The same setup as the previous experiment is used, with target code executing on the FOX11 board and with a real-time oscilloscope sampling the voltage differential across a shunt resistor in series between the board and the power supply.

The target code is similar to that of the previous experiment, with a marker followed by a routine toggling the contents of Acc A in an infinite loop. The source code is shown in Listing 3. The same code is executed on two different HC11 processors—the processors are placed in the same FOX11 board.

Listing 3. Original instructions used to detect subtle register changes

```
12          oraa    #$ff    ;a = ff
13          clra            ;a = 00
14          ldaa    #$ff    ;a = ff
15          anda    #$00    ;a = 00
16          eora    #$ff    ;a = ff
17          ldaa    #$00    ;a = 00
18          eora    #$ff    ;a = ff
19          ldaa    #$00    ;a = 00
20          oraa    #$ff    ;a = ff
21          eora    #$ff    ;a = 00$
```

Fingerprint extraction is similar to the previous experiment, with the exception that several traces are aligned and averaged in order to form the fingerprint. We analyze traces from both processors. Each trace contains two execution cycles of the target code. When we correlate a fingerprint obtained from the first processor with traces captured during the execution of the target code on the second processor, the correlation spikes show again, as shown in Figure 6.

To better visualize the similarities, we use the random variable X formed by the peak correlation values, x, for every execution instance i, given by:

$$x_i = \max_k \left\{ \rho_{ST}^{(i)}(k) \right\}$$
$$X = \{x_i\} \qquad i \in \{1, \ldots, N\}$$

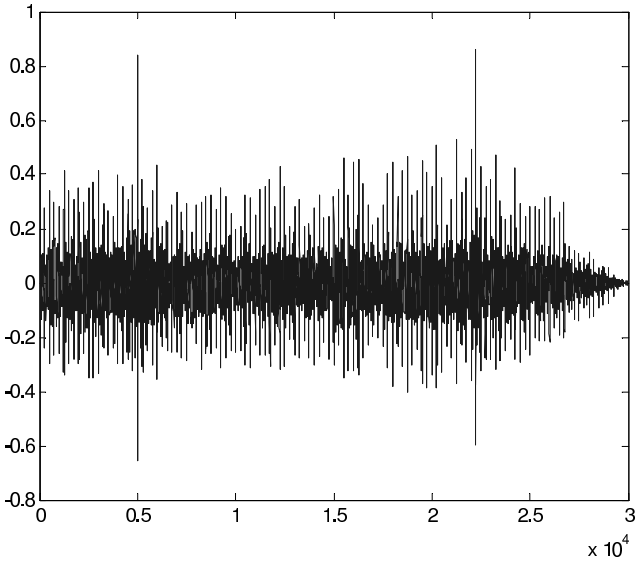

Fig. 6. Cross-Correlation with traces from a different processor

One hundred traces are collected from each processor. X_O includes the peak values resulting from correlating the signature with traces obtained from the execution on the original processor. X_A is obtained by correlating with traces from the alternative processor. The sample distribution of both scenarios is shown in Figure 7.

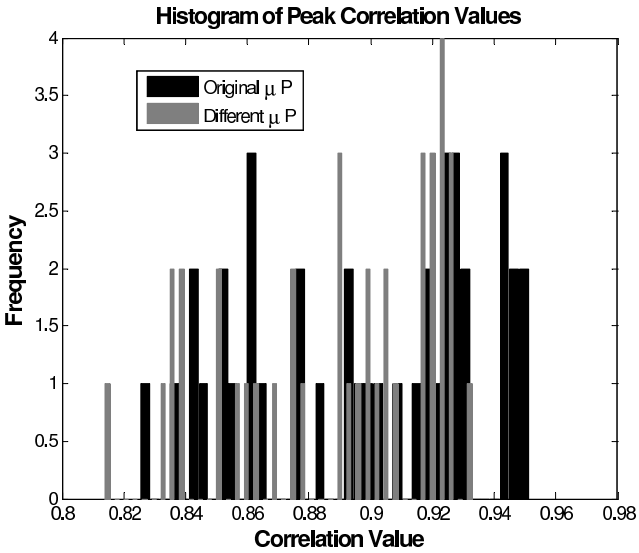

Fig. 7. Peak Correlation Values Sample Distribution

5. CONCLUSIONS

In this paper we presented a novel approach to leverage SDR and CR regulatory compliance based on dynamic power consumption monitoring. The approach delivers a runtime independent monitor which uses power fingerprints to determine whether authorized software, or specific critical modules, are executing. This approach can deliver an effective tool to shape future regulatory policy and reduce interference risks without placing excessive burdens on equipment manufacturers. It is important to mention, however, that the potential impact from this approach expands well beyond SDR and CR, as it is applicable to any digital system. For example, this approach can be applied to critical areas such as: adaptive traffic systems, power distribution systems, and health care systems.

While preliminary results support the feasibility of this approach, there is a large amount of research required to make it a practical reality. For example, it is necessary to characterize the robustness of this approach to uncertainty introduced by manufacturing variabilities and other elements on the boards, even for multicore systems. It is also necessary to establish the limitations in terms of sensing requirements as the speed of processors keeps increasing. Complex processor architectures can also have an impact on this approach along with operating systems. There are many issues that can impact the performance of this approach and may limit its application to certain systems. For those systems for which it is feasible, however, the prospect of having a run-time independent execution monitor motivates further research.

References

[1] Federal Communications Commission. Authorization and use of software defined radios. ET Docket No. 00-47, september 2001.

[2] Anant Sahai, Kristen Ann Woyach, George Atia, and Venkatesh Saligrama. A technical framework for light-hander regulation of cognitive radios. *IEEE Communications Magazine*, 47(3):96 – 103, 2009.

[3] John M. Chapin and William H. Lehr. Time-limited leases for innovative radios. In *2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN) 2007*, pages 606 – 619, 2007.

[4] N.H.E. Weste and K. Eshraghian. *Principles of CMOS VLSI Design: A Systems Perspective*. Addison-Wesley, 2nd edition, 1993.

[5] T. Laopoulos, P. Neofotistos, C. A. Kosmatopoulos, and S. Nikolaidis. Measurement of current variations for the estimation of software-related power consumption. *IEEE Transactions on Instrumentation and Measurement*, 52(4), August 2003.

[6] S. Nikolaidis, N. Kavvadias, P. Neofotistos, K. Kosmatopoulos, T. Laopoulos, and L. Bisdounis. Instrumentation setup for instruction level power modeling. Technical report, Springer-Verlag, 2002.

[7] Carlos R. Aguayo Gonzalez and Jeffrey H. Reed. Power fingerprinting in SDR and CR Integrity Assessment. In *IEEE Military Communications Conference (Milcom)*, 2009.

[8] Wytech Website. Available at: *http://www.evbplus.com/index.html*.