Identifying Counterfeit Smart Grid Devices: A Lightweight System Level Framework

Leonardo Babun, Hidayet Aksu, and A. Selcuk Uluagac
Cyber-Physical Systems Security Lab
Department of Electrical & Computer Engineering, Florida International University
10555 West Flagler St. Miami, FL 33174
Email: {lbabu002, haksu, suluagac}@fiu.edu

***Abstract*—The use of *counterfeit smart grid devices* throughout the smart grid communication infrastructure represents a real problem. Hence, monitoring and early detection of counterfeit smart grid devices is critical for protecting smart grid's components and data. To address these concerns, in this paper, we introduce a novel system level approach to identify counterfeit smart grid devices. Specifically, our approach is a configurable framework that combines system and function call tracing techniques and statistical analysis to detect counterfeit smart grid devices based on their behavioural characteristics. Moreover, we measure the efficacy of our framework with a realistic testbed that includes both resource-limited and resource-rich counterfeit devices. In total, we analyze six different counterfeit devices in our testbed. The devices communicate via an open source version of the IEC61850 protocol suite (i.e., *libiec61850*). Experimental results reveal an excellent rate on the detection of smart grid counterfeit devices. Finally, the performance analysis demonstrates that the use of the proposed framework has minimal overhead on the smart grid devices' computing resources.**

***Index Terms*—Smart Grid, counterfeit devices, cyber security, IEC61850, system calls, function calls.**

I. Introduction

Recently, a substantial effort to modernize the traditional power grid to the next generation of technology, i.e., *smart grid*, has occurred. The success of the smart grid vision depends on the integration of underlying electrical distribution infrastructure with communication networks. The information technology (IT) systems and devices attached to the smart grid must guarantee, despite any threat, the security and the integrity of the smart grid data and infrastructure. *With all its dependency upon device communications, the smart grid is highly vulnerable to any security risk stemming from devices*. Especially, the use of counterfeit devices can wreak havoc on the smart grid's critical functionalities [1]. The consequences of such counterfeit device-based attacks on the smart grid can be very severe [2] causing, for instance, major blackouts.

In this paper, we propose a configurable system-level framework to monitor and detect counterfeit devices which are performing unauthorized operations inside the smart grid architecture. Specifically, the proposed framework utilizes system and function call tracing techniques and statistical analysis to monitor counterfeit devices' behavior. In order to test our framework, we designed a representative smart grid testbed which executes essential operations conforming to the *International Electrotechnical Commission 61850* (IEC61850) [3], which is a protocol suite that defines the communication standards for electrical substation automation systems. To emulate the diversity of devices in the smart grid, the proposed testbed includes both resource-limited (devices with limited availability of computing resources, e.g., Remote Terminal Units (RTUs), Programmable Logic Controllers (PLCs)) and resource-rich (devices with more computing resources, e.g., Phasor Measurement Units (PMUs), Intelligent Electronic Devices (IEDs)) devices. The devices use open source $libiec61850$ libraries [4] to exchange smart grid time-critical messages using the GOOSE format. In our adversary model, we consider six different types of counterfeit devices with different computing resources and hardware capabilities. It also complies with the security requirements defined by the National Institute of Standards and Technology (NIST) [5] for the smart grid. Experimental results demonstrate an excellent detection rate for the counterfeit smart grid devices. Additionally, detailed performance analysis shows minimum overhead on the use of computing resources (i.e., CPU, memory, etc.) with our framework. On average, memory utilization does not increase more than 0.03% while real, system, and user time would not increase more than 230ms for even the worst case scenario (resource-limited device). These results were obtained by comparing the IEC61850 open source application metrics with and without using our framework.

The remainder of the paper is organized as follows. In section II, we present the related work. In section III, we describe the adversary model used in our work. Section IV focuses on the design of the proposed counterfeit smart grid device detection framework. In section V, we analyze and discuss experimental results. Finally, section VI concludes the paper and propose future research directions.

II. Related Work

In this section, we present the related work. Generally, researchers and cyber security analysts isolate the problem of counterfeiting to the smart grid supply chain. In [6] and [7], the authors present different approaches for the detection of fake electronic components. These works are mainly focused on hardware counterfeit detection. In [8], the authors present a network-based counterfeit device detection technique that analyzes network traffic in order to detect hardware-based counterfeit devices. Although this is an interesting concept, network dynamics (e.g. delay, etc.) can have an adverse effect on the detection mechanism. In [1] and [9], the authors define the problem of device counterfeiting as a case when a compromised or false electronic component or board is used in part or in the total process of assembling a smart grid function-critical device. In general, the topic of compromised devices has

978-1-4673-8999-0/17/$31.00 ©2017 IEEE

not been extensively studied in the literature. In most cases, researches focus on proposing anomaly detection mechanisms [10] for different types of attacks in the smart grid [11] without particularizing on the attack sources (e.g., compromised devices). In few cases, however, the behavior of the smart grid device is considered. In [12], the authors study the minimal number of compromised sensor in order to effectively manipulate a given number of smart grid states. Further, they consider the optimal PMU placement to defend against data integrity attacks. Other researchers and cyber security authorities try to find solutions to the problem of counterfeit devices being used in the smart grid [13]. Intelligent secure packaging, outbound beaconing, and better tracking systems are some of the countermeasures that are proposed to fight against counterfeiting on the supply chain side [1]. However, skilled attackers could have remote access to legitimate devices (e.g., RTUs, PMUs, IEDs, etc.) and create opportunities for tampering smart grid devices outside the smart grid supply chain. Additionally, the proposed mechanisms for protecting and monitoring the supply chain against counterfeiting are far from infallible. Reality is, counterfeit devices constitute a big problem for smart grid security. Counterfeit devices account for at least \$7.5B in lost revenue for U.S. semiconductor companies [14].

Our framework is different from other discussed solutions which, in most cases, focus on prevention. In fact, these solution are not intended to address the problem of counterfeit devices outside the supply chain. As discussed before, there are also cases where different approaches are being used for the detection of counterfeit devices and/or monitoring application behaviour. In none of these cases, the solution is intended to be applied in the smart grid domain. Additionally, in order to succeed, these solutions need to monitor constantly-changing environments like network traffic or computational systems. This constitutes a limitation in terms of system overhead and resource utilization. On the other hand, our framework has a simpler model and is lightweight in terms of system overhead while providing excellent detection rate of counterfeit smart grid devices.

III. Adversary model

In this section, we describe the adversary model used to define the threats caused by the counterfeit smart grid devices and also the way the attacks could be perpetrated.

For our analysis, we define a fake or counterfeit smart grid device (CD) (e.g., RTUs, PMUs, IEDs, etc.) as a device with some malicious function due to a counterfeited hardware or software component [15], [16]. The malicious function can change the basic operations of the original device and it could have been installed either in one of the supply chain stages or while performing software upgrade in the field.

Our adversary model considers, conforming to the NIST guidelines, three possible threats in the smart grid related to counterfeit devices [17]:

1) *Information leakage:* the counterfeit smart grid device can open additional communication channels to leak valuable smart grid information to the adversary (another untrusted insider or outsider).
2) *Measurement poisoning:* the counterfeit smart grid device can generate fake data that can be used to poison the real status of the smart grid.
3) *Store-and-send-later:* the fake device can store information in hidden files that can be recovered later by an attacker.

Based on these three well-defined threats and considering both resource-limited and resource-rich smart grid devices, we further define six different counterfeit devices as part of our adversary model. Each counterfeit device reflects a different combination of the aforementioned three smart grid threats and availability of computing resources. The first three counterfeit devices (CD_1, CD_2, and CD_3) have all limited computing resources and will impact the smart grid infrastructure by (1) leaking sensitive information, (2) allocating unauthorized amounts of memory to create fake data and poison real measurements, and (3) creating unauthorized hidden files to store critical information which are retrieved later by the attacker. The other three counterfeit devices defined in our adversary model (CD_4, CD_5, and CD_6) will perform the same type of attacks, respectively, but from devices with higher computing resource availability. We also assume that the counterfeit devices perform its malicious activity following a Poisson distribution, which allows for randomly and efficiently spacing the attacks and also constitutes a valid model to emulate the randomness of such events [18].

Consider t=$[0, T]$, the communication interval between the two smart grid devices. The probability of having an attack from a counterfeit device $CD_i \in \{CD_1, CD_2, CD_3, CD_4, CD_5, CD_6\}$ can be expressed as:

$$P_{cd} = \frac{\lambda^k e^{-\lambda}}{k!}, \quad k \in \mathbb{R}, \qquad (1)$$

where λ is the average number of attacks in the interval t and k is the total number of attacks in the same interval.

In general, the increment sequence N(t) that models the aggregate attacks is defined by:

$$P(N(s+t) - N(s)) = e^{-\lambda t} \frac{(\lambda t)^k}{!k} \quad k \in \mathbb{R}, s > 0, \qquad (2)$$

where $N(s+t) - N(s)$ is called a length t increment of the attack process $N(t) : t \geq 0$.

IV. Overview of Framework

In this section, we describe the design of the proposed framework.

A. System model with a realistic testbed

Our framework considers a realistic scenario from a smart grid substation. The testbed's configuration includes publisher-subscriber two-way communication configuration which sends and receives GOOSE messages. For this purpose, we utilize an open source version of IEC61850 [4] protocol running on Linux-based systems. *The use of open source software increases our framework's interoperability, flexibility, and opens new possibilities for customizations and for making this tool more configurable*. Our resource-limited device (GOOSE publisher) runs on a Raspberry Pi 2B, using Advance RISC Machine (ARM) 32 bits architecture with limited memory and CPU resources. The resource-rich device (GOOSE subscriber) runs on a

Linux Ubuntu 14.04 virtual machine with a more powerful CPU and higher memory capabilities (see Fig. 1). Finally, we applied two different call tracing techniques: library interposition to hook and trace system calls (at kernel level) and Process Trace (ptrace) to hook and trace function calls (at user level).

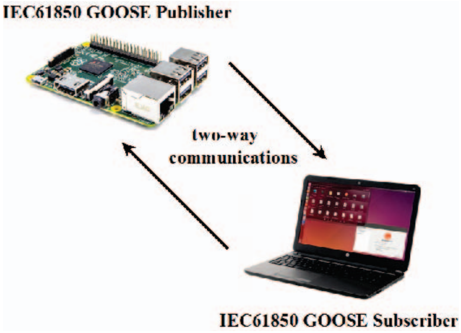

Figure 1. Smart grid testbed configuration using resource-rich and resource-limited devices conforming to IEC61850.

1) Learning Process: The application of our framework requires of the utilization of ground truth devices that can be used as a reference for correlation purposes. Basically, the proposed framework compares and correlates, based on three different detection approaches, the statistical information of system and function call lists from equivalent ground truth and unknown devices that are performing similar tasks in the smart grid infrastructure (see Fig. 2). For this purpose, we create a database of different ground truth profiles (GTP) from genuine smart grid devices throughout our learning process. In order to be used for correlation analysis, our GTPs have to be obtained from smart grid devices that perform very stably while executing normal smart grid operations. In other words, the system/function call lists obtained from the same process and at different time intervals in the same device need to be highly correlated. To calculate autocorrelation, we assign a different weight δ_i to different types of system/function calls in the order they appear. The assignment of δ_i weights can be done randomly or by following a specific assignment criteria. This criteria can depend on the importance of the system/function calls, the type of application that we are evaluating, etc. (adaptive assignment). As a result of the assignment process, we obtain a random variable X that takes values between δ_{min} and δ_{max}. Finally, we use X to calculate the *statistical-autocorrelation-simple* as described in Equation 3:

$$\rho_{X_0 X_i} = \frac{\sum x_i x_{i+t} - n\overline{x_i x_{i+t}}}{n s_{x_i} s_{x_{i+t}}}, \tag{3}$$

where x_{i+t} represents a random variable from the same random process than x_i but from different time interval t and n represents the size of the random variables.

System and function call lists normally posses some degree of randomness [19]. Particularly in the cases of resource-limited systems, this randomness can be notably higher because of the limitation on available resources during demanding system operations [20]. This limitation would stop our framework from being suitable for resource-limited smart grid devices. To overcome this, we propose the use of a *statistical-autocorrelation-advanced* technique. In this technique, we combine the values from X_i to X_{i+h} in X, resulting in a new random vector X' smaller in size and with lower random component. The index h represents the number of individual calls from the original list that we are combining to create a new random value. Hence, this index value h is proportional to the amount of randomness to be removed and constitutes another configurable parameter in our framework.

Finally, once generated, the GTPs include the following information: (1) type and amount of system/function calls (TASC) and (2) the entire system/function call list (SCL), both triggered during the learning process. The final format of the GTP looks as follows:

$$GTP = \{\mu(TASC), SCL\}, \tag{4}$$

2) Framework description: There are three fundamental operations defined in our framework (see Fig. 2):

- *Data collection*: in this step we apply the proposed system/function call tracing techniques. The objective is to trace and capture all the system/function calls raised during the time interval in which the devices from our testbed are being tested.
- *Data processing*: in this step, we apply up to three different detection approaches to extract, compare, and correlate information from the system/function call lists and the ground-truth device profiles obtained from the ground-truth profile database.
- *Decision*: finally, the decision algorithm processes the results from steps 1 and 2 to decide if the devices are genuine or counterfeit.

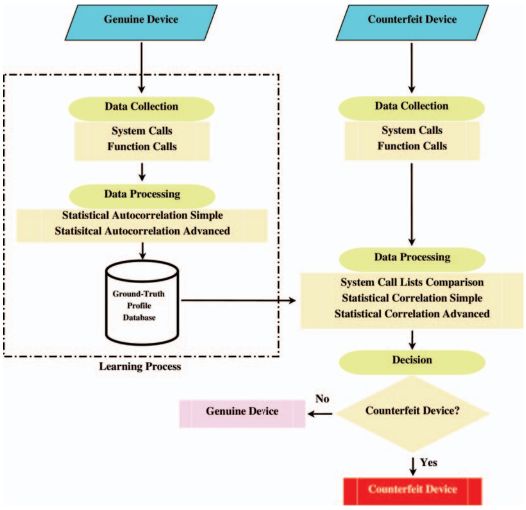

Figure 2. Configurable framework for monitoring and detecting counterfeit smart grid devices.

B. Detection approaches

Lets assume we have an unknown device that we are trying to decide as genuine or counterfeit. The following detection approaches will be applied as part of our framework.

1) $Detection - Method_1$ (Direct System Call Comparison): This detection approach will directly compare the two smart grid devices (ground truth and unknown) (Fig. 2) based on the type and frequency of system and function calls triggered during the device's interaction. In general, the execution of this approach is very light in terms of computing resources (see Section V). As part of this first approach, the average number per system/function call is calculated for the unknown device. Then, this value is normalized against the average number per system/system call of the ground truth profile respectively. As per $Detection - Method_1$, a normalized value greater than 1 means that extra system/function call activity was observed and potentially indicates the presence of a counterfeit device.

2) $Detection - Method_2$ (Statistical-correlation-simple): For stronger decision algorithms, our framework combines the previous detection approach with a second technique. This technique applies statistical correlation between system/function call lists from genuine and unknown devices. Statistical correlation will give valuable information of the mutual statistical relationship between system/function call activities from ground truth and unknown devices, based not only on the type and amount of system/function calls, but also in the order those calls are triggered in both devices. In this specific approach, after finishing the conversion of system and function call lists into random variables, we obtain two different group of variable. These two groups are: X_0 which describes the outcomes of the genuine device; and Y_{1-6} which constitute a group of random variables that describe the outcome of the unknown devices (for the particular case of this paper the six different counterfeit devices defined in our adversary model (Section III)). Finally, we define the correlation between X_0 and Y_{1-6} as:

$$\rho_{X_0 Y_i} = \frac{\sum x_0 y_i - n\overline{x_0 y_i}}{n s_{x_0} s_{y_i}}, \tag{5}$$

where n represents the size of X and Y, $\overline{x_0}$ and $\overline{y_i}$ represents the mean and s_{x_0} and s_{y_i} represent the standard deviation.

As per $Detection - Method_2$, correlation values below certain configurable threshold β mean that the system/function call lists from genuine and unknown devices are not highly correlated [18] and will potentially indicate the presence of a counterfeit device.

3) $Detection - Method_3$ (Statistical-correlation-advanced): In cases where ground truth profiles were obtained after applying statistical-autocorrelation-advanced technique, the same method has to be applied on counterfeit devices before proceeding to the decision step. As mentioned before, this detection approach will try to remove randomness from system/function call lists. The novel idea introduced here combines $cfeitV[i]$ and $cfeitV[j+h]$ from the original random variable obtained in $Detection - Method_2$. This operation will result in new random variable lists $cfeitVRd$, smaller in size, and with lower random component. Finally, the correlation value between ground truth device and unknown device is calculated.

The final step of our framework would be the execution of a decision process. In this process, the framework utilizes the results from the earlier steps and deems the device as a compromised if the correlation values of system/function call lists between genuine and unknown devices falls under a configurable decision threshold ($corrXY < \beta$).

V. Performance Analysis and Discussion

In this section, we analyze the performance of the proposed framework. In all the cases, the results are obtained after averaging 30 different runs for all covered scenarios. The scenarios include six different types of counterfeit devices based on three different threats and two different types of devices (resource-limited and resource-rich) as described in our adversary model in Section III.

A. Detection performance

Tables I and II summarize some of the system and function calls detected in the resource-limited and the resource-rich devices respectively, after using $Detection - Method_1$. In both cases, columns from 3 to 6 list the average rate of system and function calls normalized against the average rate corresponding to the number of system/function calls from the GTP. Specifically, column 3 lists the values corresponding to the genuine device and columns from 4 to 6 the values corresponding to counterfeit devices (CD_1, CD_2, CD_3 for resource-limited devices and CD_4, CD_5, CD_6 for resource-rich devices) respectively. *Any value greater than 1 (marked in gray) in columns from 4 to 6 represents extra system call activity due to the presence of counterfeit operations. That means, extra system/function calls activity reveal the presence of a counterfeit device.* It can be noticed that, by using ptrace, our framework is able to identify all cases of counterfeit devices. Successful detection was performed for both resource-rich and resource-limited smart grid devices. In the case of library interposition, only Counterfeit Devices 1 and 4 (CD_1 and CD_4) were properly detected. In the other cases, variations on system/function call rates (in CD_2, CD_3, CD_5 and CD_6) were ineligible if compared to the genuine device.

Table I
Normalized rate of system and function calls detected after using our framework on counterfeit resource-limited devices (e.g., RTUs, PLCs).

	Type of call	**Orig.**	CD_1	CD_2	CD_3
ptrace	brk	1	1	6.7	1
	$clone$	1	12.5	1	1
	$close$	1	1	1	3.2
	$fstat64$	1	~1	1	8.8
	$mmap2$	1	2.4	4.4	2.4
	$mprotect$	1	2.8	1.1	1
	$munmap$	1	1	2	13
	$open$	1	1	1	5
	$rt\_sigprocmask$	1	8.7	1	1
	$rt\_sigaction$	1	1	3	3
Interposition	$close$	1	1	1	1
	$free$	1	3.2	~1	~1
	$malloc$	1	3.3	~1	~1
	$memcpy$	1	1	1	1
	$mmap$	1	12.5	1	1
	$mprotect$	1	12.5	1	1
	$pthread\_create$	1	12.5	1	1
	$sendto$	1	4.3	~1	~1
	$signal$	1	24	1	1
	$usleep$	1	3.5	~1	~1

As we stated in our system model, the first detection method is always combined with $Detection - Method_2$ to strengthen the detection algorithm. Figure 3 shows the results after calculating the correlation between system and function call lists from genuine and counterfeit devices. For successfully applying this detection approach, we first

Table II
NORMALIZED RATE OF SYSTEM CALLS DETECTED AFTER USING OUR FRAMEWORK ON COUNTERFEIT RESOURCE-RICH DEVICES (E.G., PMUS, IEDS).

	Type of call	**Orig.**	CD_4	CD_5	CD_6
ptrace	*brk*	1	1	8.3	1
	clone	1	23	1	1
	close	1	6.5	6.8	6.75
	fstat	1	12	12.5	12.25
	mmap	1	4.1	6.64	2.6
	mprotect	1	3.4	1.1	1
	munmap	1	23	26	24
	open	1	6.5	6.75	6.8
	rt_sigaction	1	8.3	1	1
	rt_sigprocmask	1	1	1	1
Interposition	*free*	1	15.6	~1	~1
	malloc	1	15.6	~1	~1
	memcpy	1	17.8	1.1	~1
	mmap	1	24	1	1
	mprotect	1	24	1	1
	pthread_create	1	24	1	1
	pthread_detach	1	24	~1	1
	recvfrom	1	15.7	~1	~1
	signal	1	24	1	1

need to have a genuine device we can trust. A ground truth device is found after obtaining high autocorrelation values ($\rho_{x,x} > 0.6$) [18] among 30 different realizations of system/function call lists obtained from the same genuine device at different time intervals. Then, the ground truth device can be used to calculate the corresponding GTP. Correlation values between ground truth and counterfeit devices are expected to be low ($\rho_{x,y} < 0.6$) [18]. On the other hand, correlation values between ground truth and genuine devices are expected to be high ($\rho_{x,y} > 0.6$) [18].

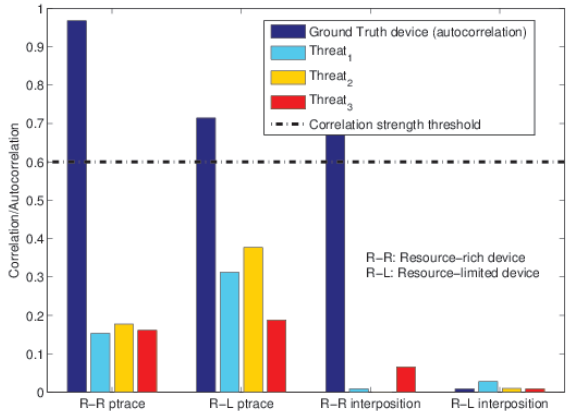

Figure 3. Correlation between ground truth and counterfeit devices for both resource-rich and resource-limited devices after applying the second detection method from our framework.

Experimental results after applying $Detection-Method_2$ can be found in Figure 3. We can observe that, by using ptrace, it is possible to obtain low correlation values (in the range of 0.15 to 0.35) between system and function call lists from genuine and counterfeit devices. By setting the correlation strength threshold at 0.6 (moderate to high correlation [18]), the framework is able to detect all cases of counterfeit device. For the case of library interposition, the framework performs very well for resource-rich counterfeit devices. However, for resource-limited counterfeit devices $Detection-Method_2$ cannot be applied as a detection approach. If we observe in Figure 3, autocorrelation values of resource-limited ground truth devices are very low when library interposition is applied. Resource-limited systems generally have higher levels of randomness during kernel execution which normally result in system/function call lists with higher random component [20]. This levels of randomness is what makes impossible to apply correlation approach in this specific case (library interposition on resource-limited device).

In order to overcome the previous limitation, we apply our third detection method which utilizes statistical-correlation-advanced techniques. Experimental results after applying $Detection-Method_3$ show an important improvement on the autocorrelation values between system call lists from resource-limited ground truth device (see Fig. 4). By using this technique, our framework was able to find a stronger GTP to be used as a reference in the detection of resource-limited counterfeit devices. Finally, correlation between system call lists obtained from the ground truth device and the resource-limited counterfeit devices was applied. In Figure 4 we are comparing the outcomes from $Detection-Method_2$ and $Detection_Method_3$ for the case of library interposition applied on resource-limited devices. As can be observed, after applying $Detection-Method_3$ the framework obtained high autocorrelation values (over 0.6) for the case of ground truth device and lower correlation values (under 0.6) after comparing the genuine device to the counterfeit device.

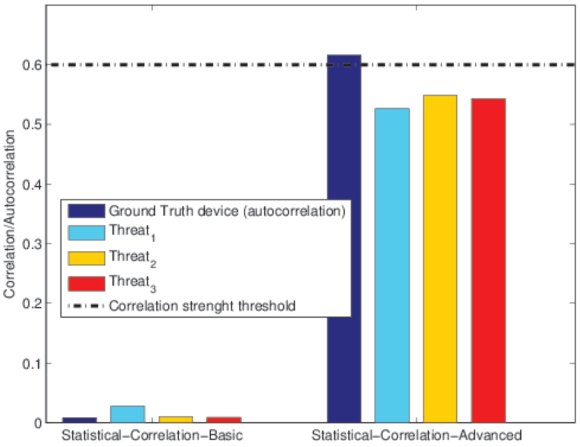

Figure 4. Statistical-Correlation-Advanced between resource-limited ground truth and counterfeit devices as a result of using the third detection method from our framework (library interposition case).

Table III
AVERAGE OF SYSTEM OVERHEAD ON RESOURCE-RICH (E.G. IEDS, PMUS) AND RESOURCE-LIMITED DEVICES (E.G. RTUS, PLCS) AS A CONSEQUENCE OF USING OUR FRAMEWORK.

Metrics	**NF**		**WF**			
	value	value	**ptrace (%)**		**LI (%)**	
	R-R	**R-L**	**R-R**	**R-L**	**R-R**	**R-L**
RT (s)	60.00	60.11	**0.05**	**3.8**	**0.01**	**0.1**
ST (s)	0.49	3.60	**8.1**	**3.6**	**10.2**	**5.5**
UT (s)	0.31	0.49	**16.1**	**0.31**	**6.4**	**2.0**
Mem (KB)	1967.5	1827.5	**1.1e-3**	**4.3e-5**	**3.0e-2**	**1.0e-3**
CPU (%)	1	6.02	**0**	**1.9**	**0**	**1**

B. System overhead

As we mentioned before, our framework has to perform with relative accuracy and scalability, but without introducing too much overhead. Table III summarizes average of system overhead on resource-limited and resource-rich devices. The metrics RT, ST, UT, Mem and CPU

correspond to the values of real time, system time, user time, memory, and CPU respectively. In this table, **NF** (No Framework) represents the case where no framework was used and **WF** (With Framework) represents the cases where we executed our experiments while using our framework. Also, **LI** represents the cases where Library Interposition was used, **R-R** refers to resource-rich devices, and finally **R-L** refers to resource-limited devices. Results demonstrate that, for resource-limited devices, even high-resource utilization techniques like library interposition do not impact performance considerably. For the particular cases of real, system, and user time, increments not greater than 230ms are observed. For critical metrics like memory and CPU, worst case scenario shows that our framework utilizes 0.03% more of memory (out of the total memory available on the device) and 1.9% more of the CPU. For resource-rich devices, the overhead introduced as a result of using our framework is even lower if compared with resource-limited devices. For the cases of real, system, and user time, we could observe increments not greater than 50ms. For critical metrics like memory and CPU, our framework utilizes 0.001% more of memory (out of the total memory available on the device) and 0% more of the CPU. In summary, for resource-limited and resource-rich devices, library interposition introduces the most overhead to the system. However, this overhead is considerably low if compared with similar applications proposed in the literature [21], [22].

VI. Conclusions and Future Work

In this work, we designed a system-level configurable framework capable of monitoring and detecting counterfeit smart grid devices. Our framework combines system and function call tracing techniques (i.e., library interposition, ptrace) and statistical analysis (simple and advanced) to monitor and detect counterfeit device behaviour. Moreover, we evaluated the performance of our framework on six different counterfeit device configurations conforming to realistic smart grid scenarios. Experimental results demonstrated that our framework is able to successfully detect different types of counterfeit device behaviour in a variety of different environments. Also, our performance analysis reveals that the use of the proposed counterfeit device detection framework does not have a significant overhead on the smart grid devices' computing resources. Our future work will focus on analyzing and measuring the performance of the framework with traditional security metrics such as accuracy, recall, precision, and precision.

VII. Acknowledgment

This material is based upon work supported by the Department of Energy under Award Number DE-OE0000779.

References

[1] D. van Opstal, U.S. Resilience Project, "Supply chain solutions for smart grid security: Building on business best practices." Sep 2012. [Online]. Available: http://usresilienceproject.org/wp-content/uploads/2014/09/report-Supply_Chain_Solutions_for_Smart_Grid_Security.pdf

[2] Y. Mo, T. Hyun-Jin Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig and B. Sinopoli, "Cyber-physical security of a smart grid infrastructure," in *Proceedings of the IEEE*, vol. 100. New York, NY, USA: IEEE, 2011, pp. 195–209.

[3] C. Kriger, S. Behardien and J. Retonda-Modiya, "A Detailed Analysis of the GOOSE Message Structure in an IEC 61850 Standard-Based Substation Automation System," *Int. Journal Comp. Comm.*, vol. 8, no. 5, pp. 708–721, Oct. 2013.

[4] M, Sillgith, "Open source library for IEC 61850: Release 0.9," Feb 2016. [Online]. Available: http://libiec61850.com/libiec61850/

[5] The smart grid interoperability panel - cyber security working group, "Introduction to NISTIR 7628: guidelines for smart grid cyber security," Sept 2010. [Online]. Available: http://www.nist.gov/smartgrid/upload/nistir-7628_total.pdf

[6] S. Sathyanarayana, W. H. Robinson, and R. Beyah, "A network-based approach to counterfeit detection." in *IEEE International Conference on Technologies for Homeland Security*, ser. HST, Waltham, Massachusetts, 2013, NS.

[7] A. Kanovsky, P. Spanik and M. Frivaldsky, "Detection of electronic counterfeit components," in *2015 16th Int. Scientific Conf. on Electric Power Engineering (EPE)*. Kouty nad Desnou: IEEE, May 2015, pp. 701 – 705.

[8] U. Guin, D. Forte, and M. Tehranipoor, "Anti-counterfeit techniques: From design to resign," in *Proceedings of the 2013 14th International Workshop on Microprocessor Test and Verification*. Washington, DC, USA: IEEE Computer Society, 2013, pp. 89–94. [Online]. Available: http://dx.doi.org/10.1109/MTV.2013.28

[9] W. E. Cobb, E. W. Garcia, M. A. Temple, R. O. Baldwin and Y. C. Kim, "Physical Layer Identification of Embedded Devices Using RF-DNA Fingerprinting," in *2010 - MILCOM 2010 Military Comm. Conf.*, San Jose, CA, Oct. 2010, pp. 2168 – 2173.

[10] A. M. Kosek, "Contextual anomaly detection for cyber-physical security in smart grids based on an artificial neural network model," in *2016 Joint Workshop on Cyber- Physical Security and Resilience in Smart Grids (CPSR-SG)*, April 2016, pp. 1–6.

[11] Y. Sun, X. Guan, T. Liu, and Y. Liu, "A cyber-physical monitoring system for attack detection in smart grid," in *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2013, pp. 33–34.

[12] Q. Yang, R. Min, D. An, W. Yu, and X. Yang, "Towards optimal pmu placement against data integrity attacks in smart grid," in *2016 Annual Conference on Information Science and Systems (CISS)*, March 2016, pp. 54–58.

[13] Z. A. Collier, D. DiMase, S. Walters, M. M. Tehranipoor, J. H. Lambert, and I. Linkov, "Cybersecurity standards: Managing risk and creating resilience," *Computer*, vol. 47, no. 9, pp. 70–76, Sept 2014.

[14] F. Koushanfar and et al., "Can EDA combat the rise of electronic counterfeiting?" in *Proc. of ACM/EDAC/IEEE Design Automation Conference*. San Fransisco, CA: IEEE, 2012, pp. 133–138.

[15] K. Huang, J. M. Carulli, and Y. Makris, "Counterfeit electronics: A rising threat in the semiconductor manufacturing industry," in *ITC. IEEE Computer Society*. IEEE, 2013, pp. 1–4.

[16] J. O'Brien and K. Lehtonen, "Counterfeit mobile devices - the duck test," in *10th Int. Conf. on Malicious and Unwanted Software (MALWARE)*. Fajardo: IEEE, 2015, pp. 144–151.

[17] N. Komninos, E. Philippou and A. Pitsillides, "Survey in smart grid and smart home security: issues, challenges and countermeasures," *IEEE Communications Surveys and Tutorials*, vol. 16, pp. 1933–1954, 2014.

[18] S. M. Ross, *Probability Models for Computer Science*, 1st ed. Orlando, FL, USA: Academic Press, Inc., 2001.

[19] E. Eskin, W. Lee and S. J, Stolfo, "Modeling System Calls for Intrusion Detection with Dynamic Window Sizes," in *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX '01*. Anaheim, CA: IEEE, Jun. 2001, pp. 165–171.

[20] H. Corrigan-Gibbs and S. Jana, "Recommendations for randomness in the operating system or, how to keep evil children out of your pool and other random facts," in *Proceedings of the 15th USENIX Conference on Hot Topics in Operating Systems*, ser. HOTOS'15. Berkeley, CA, USA: USENIX Association, 2015, pp. 25–25. [Online]. Available: http://dl.acm.org/citation.cfm?id=2831090.2831115

[21] R. Sekar, M. Bendre, D. Dhurjati, and P. Bollineni, "A fast automaton-based method for detecting anomalous program behaviors," in *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, ser. SP '01. Washington, DC, USA: IEEE Computer Society, 2001, pp. 144–. [Online]. Available: http://dl.acm.org/citation.cfm?id=882495.884433

[22] H. H. Feng, O. M. Kolesnikov, P. Fogla, W. Lee, and W. Gong, "Anomaly detection using call stack information," in *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, ser. SP '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 62–. [Online]. Available: http://dl.acm.org/citation.cfm?id=829515.830554