Towards a Framework for Evaluating the Security of Physical-Layer Identification Systems

Ryan M. Gerdes[1], Mani Mina[2], and Thomas E. Daniels[2]

[1] Utah State University, Logan, UT 84322, USA
ryan.gerdes@usu.edu
[2] Iowa State University, Ames, IA 50011, USA
{mmina,daniels}@iastate.edu

Summary. In recent years researchers have shown that the analogue signalling behaviour of digital devices can be used for identification and monitoring purposes. The basic postulate of these so-called physical-layer identification (PLI) approaches is that devices are sufficiently variable in their behaviour to be distinguishable and that an attacker would be unable to adequately emulate this behaviour. Recent work, however, has shown that at least some PLI implementations can be defeated using electronic equipment capable of generating arbitrarily shaped signals known as arbitrary waveform generators (AWGs).

In this work we first present a framework to determine whether an AWG, specified in terms of resolution, sampling rate, distortion, and noise parameters, could be used to defeat a given PLI system. We then utilise this framework in the formulation of a cost-minimisation problem to find the most cost-effective values of these parameters; i.e. we characterise the least expensive, and hence lowest performing, AWG an attacker would require to defeat a PLI system. The use of the framework is illustrated by applying it to a previously proposed PLI approach. Results indicate that the PLI system could be defeated using an AWG with a substantially lower sampling rate and resolution than the PLI system sampler.

1 Introduction

Identifying digital devices based on signalling differences manifested at the physical layer (known as physical-layer identification or PLI) has been shown to be effective for a wide range of technologies. From wired [1] and wireless [2–6] networking devices to sensor [7, 8] and RFID devices [9–11], PLI approaches are able to reliably distinguish between highly similar devices with accuracies of over 90% [12, 13].

The methodology of PLI is similar to that of biometrics [14]: (1) identify and acquire a recurring and ubiquitous signal, $\mathcal{S}$, to serve as a 'fingerprint', (2) extract a set of features from the signal, $L = f(\mathcal{S})$, and (3) employ a classification technique to compare a test feature set with a database of existing feature sets in order to verify the purported identity of the test subject. When a threshold technique is used in (3) to compare feature sets, a reference feature set, L_R, is

A.D. Keromytis and R. Di Pietro (Eds.): SecureComm 2012, LNICST 106, pp. 328–348, 2013.

used with a distance measure, $d(\cdot)$, to check whether the differences between L and L_R are within a certain threshold, $d(L_R, L) \leq th$.

While PLI could be used to corroborate higher layer mechanisms used for authentication, intrusion detection, and forensics, its use in these areas is predicated on the belief that the slight variations in the signalling behaviour of devices are difficult, if not impossible, to control and duplicate. In light of recent work by Danev *et al.* [15] and Edman and Yener [16], which showed that wireless signals can be successfully forged using arbitrary waveform generators (AWGs), it is no longer possible to merely assert the inherent unreproducibility of signals. Instead, we now require a framework to not only judge the security of PLI systems, in absolute terms and in relation to each other, with respect to existing AWGs but one that also specifies the performance an AWG of the future would need to defeat a given PLI system.

1.1 Paper Contributions and Structure

While existing work has shown that certain PLI systems can be defeated using AWGs, ours is the first work to consider the problem of whether a specific AWG can defeat an arbitrary PLI system. In what follows we propose, and provide implementation details of, a general framework for determining whether an AWG, characterised by sampling rate, resolution, signal-to-noise ration, and total harmonic distortion, could produce a forged signal that would be accepted by a given PLI system. By estimating the cost associated with an increase or decrease in each parameter, we can also find the least expensive—i.e. lowest performing—AWG necessary to defeat the PLI system.

As a result of this work, researchers and designers of PLI systems will be able to 1) determine if a PLI system is secure from an attacker using a given AWG; 2) compare and evaluate the relative security of systems; 3) investigate the strengths and weaknesses of different PLI methodologies to decide which features and comparison techniques are most effective in securely identifying devices; and 4) evaluate the trade-offs associated with selecting higher or lower performing equipment for acquiring device signals.

In the next subsection we provide an overview of the two works that motivated our research: we discuss which PLI systems were attacked, the equipment used, and the authors' results. In Section 2 we describe two ways in which PLI systems can be subverted, define our threat model, and note the most relevant parameters used to characterise AWGs. The modelling of the attacker's AWG is detailed in Section 3, where we also discuss how a cost minimisation problem can be defined that utilises the model to determine the most cost-effective values for the AWG performance parameters. In Section 4 we demonstrate the use of the framework by analysing the matched filter PLI system outlined by Gerdes *et al.* in [1, 12].

1.2 Related Work

In both [15] and [16] two types of attacks were carried out against the PLI approach (which utilised the demodulation characteristics of 802.11b signals) proposed by Brick *et al.* [5]; in addition, a transient-based PLI approach for

sensor nodes proposed by Danev and Capkun [8] was also examined in [15]. The PLI system of [5] was compromised in both works by creating signals with the features of known devices and through the replay of observed frames. For the former attack, false-accept rates (FAR) of 98% and 75% were reported for [15,16], respectively; in the latter attack, the FAR for [16] was 55% while the replay attack met with similar success as the generation attack for [15]. The difference in attack success rates can probably be attributed to not only the threat models but the vastly different hardware used to implement the PLI system and carry out the attacks.

In [15] universal software radio peripherals (USRP) operating at 128 Megasamples/s and controlled with the GNU Radio library were used for both the genuine and attacker devices, with the attacker device being programmed to produce the features of the genuine devices as measured by, and at, the PLI system (which consisted of an Agilent Digital Signal Analyser operating at 40Gigasamples/s with 8000MHz of bandwidth). The replay attack was carried out using a Tektronix AWG 7000 (20 Gigasamples/s); the frames used for the replay were captured at the attacker's location using the PLI system. In [16] both the PLI system and the attacking device were built using the same USRP (14-bit analogue-to-digital converter operating at 100 Megasamples/s and dual 16-bit digital-to-analogue converter operating at 400MHz). The attacker sought to reproduce or generate signals, which it captured, from one of three laptops used to represent legitimate users.

In their analysis of the PLI system of [8], Danev *et al.* were able to successfully replay frames captured by the PLI system over a wired channel; however, when a wireless channel was used the system could only be defeated if the attacker assumed the genuine device's physical location.

2 Preliminaries

The following notation and nomenclature will be used when discussing the analogue signalling behaviour of digital devices and the PLI system used to identify those devices. We will also assume that devices transmit data using frames, as in IEEE 802.3 and 802.11b.

A record, r, is defined as a discrete time/voltage sampled version (obtained using an analogue-to-digital converter) of the analogue signal that makes-up the data frame. For the PLI system, L_i^j is used to represent the feature vector derived from the j^{th} frame of the i^{th} device; $L_i^j(k)$ denotes access to the k^{th} element of the feature vector. In addition, a collection of feature vectors from the i^{th} device are denoted as $\mathbf{L}_i$, where $\mathbf{L}_i(j)$ is used to refer to the individual vector L_i^j. The feature vectors of frames that are to be tested by the system are always accompanied by the subscript T; the reference feature vector(s) used to establish a device's baseline behaviour by the subscript R. A generic analogue signal is denoted by $\tilde{s}$ and a sampled version of it s.

2.1 Attack Types

We define and discuss the two classes of attacks that can be used against PLI systems.

Type I Attack. We define a type one attack as an attack in which an attacker is attempting to accurately reproduce those portions of a device's signal used for identification. In the terminology of [15], this type of attack can be carried out through *feature replay* or *signal replay*. In the former case, the attacker attempts to replicate only the specific features used by the PLI system for identification; those portions of the signal not used for identification needn't be considered. For a signal replay attack, the attacker acquires a sampled version of a device's signal and attempts to produce near-perfect copies of those portions of the signal used for identification using an AWG.

We provide a demonstration of how our framework can be used to measure the resiliency of PLI systems to signal replay attacks in Section 4. Because the PLI system analysed in that section uses each sample point of the device's signal as features, the feature replay attack is not examined in this work. We note that the framework can be used to evaluate feature replay, though.

Type II Attack. In a type two attack the attacker does not seek to produce a high-fidelity copy of a device's signal but rather exploits the limitations of the identification technique used by the PLI system. For example, consider a PLI system using a threshold-based approach where the distance measure is simply the sum of the differences between the test and reference feature vectors ($d(L_R, L_T) = L_R(1) - L_T(1) + \ldots + L_R(n) - L_T(n)$, with $d(L_R, L_T) \leq th$ for L_T to be accepted). To defeat the PLI system the individual differences between all the elements of the feature vectors needn't be sufficiently small, only the sum of the differences; thus an attacker could simply engineer a signal such that $L_T(n) \geq th - L_R(1) + L_T(1) - \ldots - L_R(n)$ to satisfy the threshold.

A type two attack could be effected through manipulation of a signal generated by a device under the attacker's control or the attacker could craft a signal using an AWG. The only limitation faced by the attacker is that their signal must behave according to the standard governing data transmission for the device (for example, in the case of 10Mb Ethernet the voltage levels, signal transitions, etc must be in accordance with those specified in the 802.3 standard [17]).

To carry out such an attack, however, requires more knowledge of the PLI system and associated target device than a type one attack. Whereas a type one attack can be carried out simply by observing frames from the targeted device, in a type two attack, assuming a threshold scheme is used by the PLI system, the attacker must possess both the device's reference feature set and thresholds for future outputs to be able to construct their signal. By knowing these along with the distance measure, an attacker might be able manipulate their signal, in whole or in part, to produce a signal falling within the threshold for the device. We are aware of no attacks of this type having been demonstrated against PLI systems.

While this type of attack is not amenable to a general analysis, due to the complicated and PLI-specific relationship between the signal, feature vectors, and distance function, so long as an AWG is used to actually generate a specially crafted signal our framework can be used to determine if the attack would succeed for a given AWG. A type two attack is proposed and evaluated in Section 4.

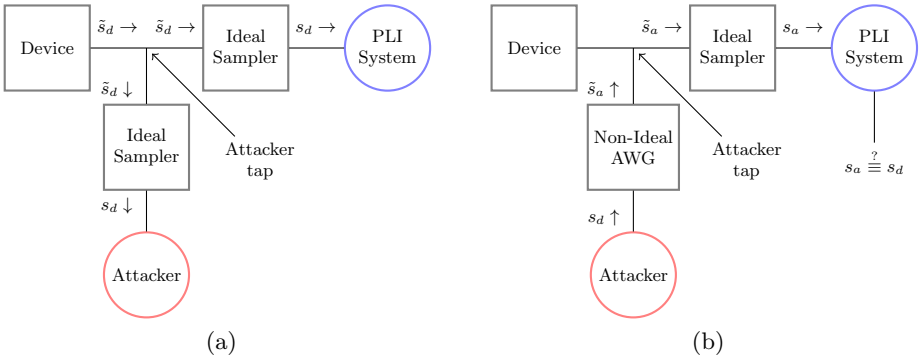

Fig. 1. (Threat model) Assuming lossless channel (a) attacker and PLI system, using the same samplers, are able to measure device's signal $\tilde{s}_d$ and obtain same sampled version s_d, and (b) attacker uses a non-ideal AWG to synthesise the analogue signal $\tilde{s}_a$ from s_d, and the PLI system, using the same sampler as in (a), measures $\tilde{s}_a$ to determine whether the attacker's signal is distinguishable from s_d

2.2 Threat Model

To simplify our analysis we chose to ignore channel effects and equip both the attacker and PLI system with ideal samplers of the same resolution and sampling rate (these parameters are explained below). In consequence of these assumptions, an attacker would be able to 1) capture the same device signal, $\tilde{s}_d$, as the PLI system (Figure 1a), and 2) generate a forged version of $\tilde{s}_d$, denoted by $\tilde{s}_a$, using an AWG and know it be identical to what will be measured at the PLI system, taking into account differences in the sample rates and resolutions of the AWG and PLI system sampler, to produce the sampled signal s_a (Figure 1b).

To justify ignoring channel effects at this time, despite the very real obstacle they present to an attacker, as demonstrated in [13], we note that a non-ideal channel is not only a problem for an attacker. Simply changing a device's position with respect to the PLI system significantly degrades our ability to re-identify it (unless training data has been previously acquired for the new position) [8]. Our analysis thus presents a best-case scenario for the attacker. In actuality an attacker would be required to model the channel and integrate its effect into the signal to be produced by the AWG.

The decision to provide the attacker and PLI system with identical samplers was mostly a practical matter: doing otherwise would have required multiple oscilloscopes to carry out our experiments. It is also difficult to see the benefit of an attacker using a sampler with a higher resolution and sampling rate than the PLI system as, irrespective of the sampling rate and resolution of the attacker's AWG, the forged signal would be downsampled at the PLI system. In addition, we would argue, and indeed it is assumed in our AWG framework, that an attacker captures $\tilde{s}_d$ at a resolution and sampling rate greater than or equal to that of their AWG for the simple reason that upsampling the captured signal could add no new information. Both of these cases could be tested at a future

Fig. 2. (Arbitrary waveform generator) the code specifies the levels of discrete signal $\hat{x}$, which the low-pass filter smooths to create $\tilde{x}$ (NOTE: $\hat{x}$ has discrete levels but is continuous in time)

time, and the framework is, in any case, flexible enough as-is to accommodate different samplers for the attacker and PLI system.

We also note that while both the channel and tap of Figure 1 are depicted using lines, this is not meant to imply that the framework is limited to analysing wired PLI systems. In the case of a wireless channel, an antenna would serve as the tap and if down-mixing were used by the PLI system (as in [8]) appropriate mixers could be placed in front of the ideal samplers and after the non-ideal AWG. If down-mixing were not used by the wireless PLI system, we could either stipulate that the sampling rate of the non-ideal AWG be no less than twice the carrier frequency used by the devices or place a mixer after the AWG to up-mix the generated signal.

2.3 AWG Characteristics

An arbitrary waveform generator creates an analogue version of a digitized waveform. The three core components of an AWG are the waveform source memory, digital-to-analogue converter (DAC), and low-pass filter (Figure 2); optional components include scaling circuits, DC offset circuits, and differential outputs [18]. An analogue signal is created by feeding the binary values of the digitized waveform (known as codes) to the DAC, where a stepped, analogue output is generated; the stepped output is smoothed by the low-pass filter.

Because of the central role of the DAC in recreating the digital signal, we will concentrate our performance analysis exclusively on it and assume the other components of the AWG to be ideal. In any case, the parameters related to the DAC we will be discussing are always given with respect to the output of the AWG, so we are merely overestimating the minimum performance of the AWG.

According to [19], the most important specifications used to evaluate the dynamic performance of a DAC are settling time, glitch impulse area, distortion, spurious free dynamic range (SFDR), and signal-to-noise ratio (SNR). In addition to these parameters, we will also discuss DAC resolution. Definitions for each of these parameters may be found in the appendix. Static performance measures (gain, offset, differential non-linearity [DNL], and integral non-linearity [INL], see [20]) are not discussed due to the fact that dynamic non-linearities dominate at high frequencies [21]. Our distortion model does, however, allow us incorporate errors due to static non-linearities.

3 Framework Overview

A system diagram of our framework is given in Figure 3. The attacker begins with s_d, a sampled version of some authenticated device's signal, $\tilde{s}_d$, that is acquired at the PLI systems sampling rate, f_p, and resolution, R_p. Because of

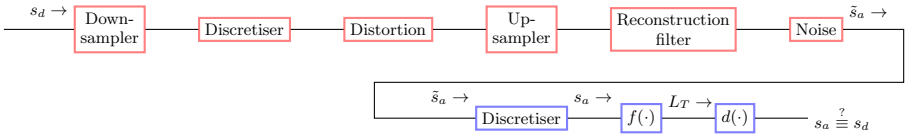

Fig. 3. (Framework overview) Operations used to simulate (red) attacker producing analogue signal $\tilde{s}_a$ from authentic device's sampled signal s_d, and (blue) PLI system sampling and comparing attacker's forged signal to baseline behaviour of device to determine whether authentic and forged signal are distinguishable

the lossless channel assumed in our threat model, s_d is the same for the attacker and the PLI system. The first step the attacker takes is to downsample s_d to the sampling rate of their AWG. Allowing $f_a = P/Q \times f_p$ $(P/Q \leq 1)$ to be the sampling rate of the AWG, s_d is downsampled by P/Q. The downsampled signal is then discretised according to the resolution of the AWG, R_a. To simulate the distortion and noise present in all real-world AWGs, the downsampled signal must be passed through a distortion function and have noise (in our case, additive white gaussian noise) added to the resulting signal to produce the attacker's output, $\tilde{s}_a$. However, before noise is added to the distorted signal, it is upsampled to the PLI system rate—i.e. it is upsampled by Q/P—and a reconstruction filter is applied. Upsampling at this point is done for two reasons.

In the first place, distortion and noise measurements of actual AWGs are made after the generated signal has passed through a reconstruction (low pass) filter. Applying our distortion model to an upsampled signal would introduce high frequency distortion components that would otherwise be filtered by the AWG's reconstruction filter. Secondly, since we are synthesising signals for the PLI system to compare with actual sampled data to determine the similarity between the two, the synthesised data must be at the same sample rate as the original. In actuality the attacker's AWG would produce a continuous-time signal that would then be sampled by the PLI system at the rate f_p; upsampling the discrete representation of the attacker's signal simulates this sampling.

At the PLI system, $\tilde{s}_a$ is discretised according to the sampler resolution R_p (the sampling of the signal having been accomplished by the AWG model).

The preceding involves only the first step of the PLI methodology; steps two and three, wherein the attacker's signal is subjected to feature extraction and comparison, are specific to the PLI system under examination.

The methodology used to model the attacker's AWG and the PLI system is detailed in the next subsection, while a cost-based method for determining the most economical values for the parameters outlined in Section 2.3 for the AWG are covered in Section 3.2.

3.1 AWG and Sampler Models

The functionality of the AWG and sampler models of the framework are explained within the context of the performance parameters given in Section 2.3. Note: the text in parenthesis immediately following each parameter indicates which aspect of the framework (with reference to Figure 3) the parameter bears upon.

Settling Time. (*Down-sampler*, *Up-sampler*, *Reconstruction filter*) It is the settling time of the DAC used in the AWG that sets the ultimate limit on the maximum sampling rate of the AWG. Allowing τ to denote the settling time of the DAC, the sampling rate of the AWG, f, must be less than or equal to the inverse of the settling time ($f \leq 1/\tau$). If we stipulate that the settling time of the attacker's AWG, τ_a is much less than the inverse of the sampling rate, f_p, of the PLI system sampler ($\tau_a << 1/f_p$) the settling time may be ignored as it unlikely that the attacker's signal would be sampled during the transition period (modern AWGs are capable of meeting this requirement, see [22]). By this assumption, the glitch area may be similarly ignored.

Based on the above, we need focus only on the sampling rate of the AWG and PLI system sampler. To simulate the attacker downsampling the signal s_d by integer amounts—i.e. the new sample rate is given by $1/n \times f_p$, where n is an integer—we can simply discard every n^{th} data point; however, to downsample by a non-integer factor, of say P/Q requires upsampling (insertion of P zeros between data points), application of an anti-alias filter, and downsampling (discarding every Q datapoints) [23]. An FIR least-squared filter with a cutoff frequency of $P/Q * f_p/2$ (the Nyquist frequency) is used as the anti-aliasing filter in our implementation. The Nyquist frequency of the attacker's AWG was chosen as most commercially available DACs are able to generate signals up to their own Nyquist frequency [24].

The same procedure is used to restore the attacker's signal to the PLI system sample rate (the signal is upsampled by Q/P).

Resolution. (*Discretisers*) Because of the filtering used to downsample and upsample signals, the sample points of the resampled signals will not be exact multiples of the increment voltage of either the attacker's AWG or the PLI system's sampler. In order to incorporate the effects of the finite resolution of the AWG and sampler, it is therefore necessary to discretise these signals by rounding each sample to the nearest multiple of the increment voltage. (Algorithm 1 details how the sampled signal s is discretised for an n-bit AWG/sampler with full-scale voltage V_{FS}.)

Algorithm 1. Set resolution

Input : $\mathbf{s}$, n, and V_{FS}
Output: $\mathbf{s}^*$ (n-bit representation of $\mathbf{s_d}$)
foreach $s_i \in \mathbf{s}$ **do**
  $i \leftarrow \arg\min_m \left(\left| s_i - m \frac{V_{FS}}{2^n - 1} \right| \right)$; `//`m `is an integer`
  $\mathbf{s}^* \leftarrow \mathbf{s}^* \cup \left(i \times \frac{V_{FS}}{(2^n - 1)} \right)$;
end

Distortion. (*Distortion*) A full and proper accounting of how the output of a DAC deviates from its ideal output depends not only on the behaviour of the non-ideal components used to construct the DAC [25] but also on its architecture [26,27]. As such, it is not possible to utilise a single distortion model in our

framework. Rather, an attacker would need to select (based upon market availability or the manufacturing resources at their disposal) a distortion model for the DAC used in their AWG. While several so-called behavioural models have been proposed for many different DAC architectures and deployments [28–32], to simply illustrate how distortion models can be used in our framework we have selected a model that, while not tied to any particular architecture, nonetheless produces adjustable amounts of static and dynamic distortion.

Allowing $s[i]$ to denote the i^{th} sample point of the sampled signal s and s^* the distorted version of s, distortion of both types can be introduced using the polynomial [33]

$$\begin{aligned} s^*[i] = D(s[i]) = \beta + \alpha s[i] + \gamma \times (\beta + \alpha s[i])^2 + \delta \times (\beta + \alpha s[i])^3 \\ + \eta \times (\beta + \alpha \times (s[i] - s[i-1]))^2 + \kappa \times (\beta + \alpha \times (s[i] - s[i-1]))^3 \end{aligned} \quad (1)$$

In (1), static distortion is generated through the scaling of individual sample points, while dynamic distortion is introduced by taking the difference between two sample points.

To achieve a certain amount of distortion using this model one would create a test signal (see the [34]), apply (1) to it, and vary the coefficients until the desired THD was reached. Unfortunately, we are aware of no set procedure for how the coefficients should be modified. In the absence of formal guidelines, we follow [33] and set the initial values of the coefficients to $\alpha = 1, \beta = 0$ (no gain or offset error, as these can be compensated for), $\gamma = 0.003, \delta = 0.0001, \eta = 0.0001, \kappa = 0.002$ and vary each coefficient (excepting α and β) by a constant multiple, m, to achieve a specified distortion. Our distortion model is then

$$\begin{aligned} s^*[i] = D(s[i], m) = \beta + \alpha s[i] + m \times \gamma \times (\beta + \alpha s[i])^2 \\ + m \times \delta \times (\beta + \alpha s[i])^3 + m \times \eta \times (\beta + \alpha \times (s[i] - s[i-1]))^2 \\ + m \times \kappa \times (\beta + \alpha \times (s[i] - s[i-1]))^3 \end{aligned} \quad (2)$$

In our framework the THD of the AWG is established using a procedure similar to that of real AWGs: Equation 2 is applied to a test signal[1], consisting of a single period of a 10 MHz sine wave, sampled at the sample rate of the AWG, and m varied until the THD equals the value specified. Common test signals used in real world measurements for several DACs we examined were 1,2,4,5, and 10 MHz (see [35], e.g.). A 10 MHz test signal was selected due to the fact that the PLI system used to illustrate our framework extracts features from a 5 MHz square wave and 10 MHz sits between the fundamental frequency and the first harmonic of 15 MHz (see Section 4.3). As noted at the beginning of Section 3, the test signal is upsampled before the distortion measurements are made.

Having found an m that produces the specified THD, (2), is applied to the attacker's signal and the resulting distorted signal upsampled by Q/P to the PLI system sample rate (Algorithm 2).

[1] The attacker's signal is not used with the model to establish the THD of the AWG because it is composed of multiple frequencies, and while the THD can be calculated for any particular frequency over the bandwidth of the signal, we cannot say which particular THD represents the THD of the AWG.

Algorithm 2. Set distortion

Input : $\mathbf{s}, P, Q$, and thd
Output: $\mathbf{s}^*$ (distorted version of $\mathbf{s}$)
$\mathbf{s_t} \leftarrow$ create test signal;
$\mathbf{s_D} \leftarrow \mathbf{s_t}$; //distorted test signal
$m \leftarrow 1$;
//$THD(\cdot)$ calculates THD using Equation 2 of [34]
while $THD(\mathbf{s_t}, \mathbf{s_D}) \neq thd$ **do**
  if $THD(\mathbf{s_t}, \mathbf{s_D}) > thd$ **then** decrease m;
  else increase m;
  foreach $s_i \in \mathbf{s_t}$ **do**
    $\mathbf{s_D} \leftarrow \mathbf{s_D} \cup D(s_i, m)$;
  end
  $\mathbf{s_D} \leftarrow upsample\,(\mathbf{s_D}, Q, P)$;
end
foreach $s_i \in \mathbf{s}$ **do**
  $\mathbf{s}^* \leftarrow \mathbf{s}^* \cup D(s_i, m)$;
end
$\mathbf{s}^* \leftarrow upsample\,(\mathbf{s}^*, Q, P)$;

Spurious Free Dynamic Range (SFDR). The distortion model described above only allows one or the other of THD/SFDR to be specified (the other may be calculated). We chose to specify THD as it more informative, in the sense that the SFDR may remain constant while harmonic distortion continues to increase.

Noise. (*Noise*) Just as is the case for distortion, there are several ways to model the noise performance of DACs [36,37]. Again, for the purposes of illustration, we have selected a simple, non-behavioural model that uses additive white Gaussian noise (AWGN) for the attacker's AWG.

As noted in [34], the signal-to-noise ratio of an AWG is calculated in such a way as to exclude the effects of distortion. Therefore, we use the signal produced by the distortion model in the numerator of the SNR ratio (see Equation 1 of [34]); i.e. a distorted signal, s, produced using (2), is defined as being free of noise. Having calculated the power of this signal, $p_s = P(s)$, to achieve a specified signal-to-noise ratio, snr, we need merely generate a noise signal, s_n of equal length with power $P(s_n) = p_s/snr$ and add the two to produce a signal with both distortion and noise, $s^* = s + s_n$ (Algorithm 3).

Algorithm 3. Set SNR

Input : $\mathbf{s}$ and snr
Output: $\mathbf{s}^*$ (noisy version of $\mathbf{s}$, with SNR of snr)
$p_s \leftarrow P(s)$; //$P(\cdot)$ calculates power
$p_n \leftarrow p_s/snr$;
$s_n \leftarrow$ create signal of white Gaussian noise, having power p_n;
$s^* \leftarrow s + s_n$;

3.2 Finding Minimum AWG Performance

By following the procedure outlined above, it is possible to simulate an attacker generating a forgery of an authenticated device's signal using an AWG of a specified sample rate, resolution, THD, and SNR. This forged signal can then be used in steps two and three of the PLI methodology (feature extraction and comparison) to determine whether the attacker's AWG is sufficient to defeat a given PLI system.

To judge the security of any particular PLI system, one could of course gather performance information on all the AWGs currently available, construct AWG models for each, and simulate attacker signals. To evaluate the relative security of different systems a similar process would be followed for each, with the system that required the most expensive AWG necessary to defeat it adjudged the most secure. Consider, however, a PLI system for which no existing AWG is capable of defeating. While, through trial and error, the framework could be used to find a number of AWGs that would defeat the system, if we wished to actually manufacture such an AWG, how would we decide which combination of performance parameters would be cost-effective?

This is to say, given two theoretical AWGs capable of defeating a particular PLI, the same in every respect except that one has five bits of resolution and a THD of -90 dBc while the other has a resolution of six bits and a THD of -70 dBc, the attacker would want to select the cheaper of the two to manufacture. Finding the most cost-effective AWG may be accomplished by utilising the above framework in the constraint function of a cost-minimisation (constrained optimisation) problem that accounts for the marginal cost for improvements in each performance parameter. Such a formulation would also be useful in the case where a wide enough gap exists between the cost of manufacturing an AWG capable of defeating the system and simply purchasing an existing AWG that is known to be able to defeat it. Similarly, we would need to know the lowest performing theoretical AWG necessary to defeat a system to be able to say that the system is secure against attacks using AWGs with sample rates, resolutions, THDs, and SNRs below a certain level.

Cost Minimisation Formulation. The cost, or objective, function in our formulation, $f_c(f, n, snr, thd)$, returns the cost necessary to obtain an AWG with a sampling rate of f, resolution of n, SNR of snr, and THD of thd. Allow $s_a = AWG(s_d, f, n, snr, thd)$ to be the attacker's forgery of an authenticated device's signal, s_d, produced using the AWG with the aforementioned parameters. Furthermore, let $th = d(L_R, f(s_T))$ be the maximum distance allowed between a signal, s_T, claiming to originate from the device and the device's feature set, L_R, where the function $f(\cdot)$ extracts features specific to the PLI approach from the sampled signal s_T and $d(\cdot)$ is the distance measure the approach employs. Our minimisation problem is then

$$\min_{f,n,snr,thd} f_c(f, n, snr, thd) \text{ subject to } d(L_R, f(s_a)) \leq th \tag{3}$$

The derivation of a sample cost function is covered in Section 4.4.

Equation 3 describes a mixed-integer non-linear programming problem, with black box constraints. To ease the process of solving of it, we can impose upper and lower boundaries on each parameter, in addition to stipulating integer values for each.

Given the assumptions of our threat model, the upper bounds for the sampling rate and resolution must be those of the PLI system sampler. Modern DACs are capable of achieving THDs < -80 dBc [35] and SNRs > 75 dB [38], so our theoretical DAC must be capable of exceeding at least these numbers. Lower bounds are calculated using the framework by setting all parameters to their upper values and then choosing one parameter to decrease until the signal generated by the AWG model violates the constraint of (3); the value at which the constraint is violated is then the lower bound for that parameter. This process is repeated for each parameter. Lower bounds are thus specific to the PLI system under consideration.

To convert (3) to an integer non-linear problem, we mandate that *thd* and *snr* be integers (n is already an integer), while for the sampling rate we define f_a to be some fraction P/Q of the PLI system sampling rate (where P and Q are integers, passed separately to the optimiser). As the signal the attacker is attempting to forge is sampled at the PLI system sampler rate, and our upsampling/downsampling routine will first upsample by P and then downsample by Q, the attacker's effective sample rate would be $P/Q \times f_p$.

4 Framework Application

We demonstrate the use of the framework on the PLI approach of Gerdes *et al.*, which was proposed to identify wired Ethernet devices. In what follows we provide a brief overview of their PLI approach, describe our implementation of it, and detail how the framework was used to analyse the security of it.

4.1 Overview of PLI Approach

Using the nomenclature of Section 2 and the generic PLI methodology of Section 1, the PLI approach of Gerdes *et al.* is to [12]: (1) capture the beginning of a 10Mb Ethernet frame, known as the synchronisation signal, where a slope-based trigger is used by the sampler to detect the beginning of the frame, (2) extract a specified number of contiguous sample points, using the triggering sample point as a reference for which sample point to start with, and (3) check if the inner product of the extracted features and reference features lies between the two thresholds established for the device.

More explicitly, as laid out in Sections 4.2–3 of [12], for device k to be accepted as device i the inner product between the reference features, L_{Ri}, of the i^{th} device and the features, L^j_{Tk}, extracted from the j^{th} record, r^j_k, of the k^{th} device must fall between the thresholds th_{+i} and th_{-i}.

The reference feature vector, derived from an arbitrary record, r^l_i, of the i^{th} device is $L_{Ri} = r^l_i[trg^l_i + m : trg^l_i + n]$, where trg^l_i is the sample point in the record r^l_i at which the scope triggered and m and n are the first and last sample points, relative to the trigger, of the span of sample points used as the feature set for the

device. To account for triggering error and slight deviations in signal levels, the test feature set is actually taken to be $L^j_{Tk} = f(r^j_{Tk}, trg^j_k) = r^j_{Tk}[trg^j_k + m - \delta : trg^j_k + n + \delta]$, where δ is the number of extra sample points to include in the feature vector.

Stating the preceding in terms of a constraint equation, we have that for a record from device k to be identified by the PLI system as having originated from device i, it must satisfy

$$th_{-i} \leq \max\left(\sum_{h=1}^{n-m} L_{Ri}[h] \times L^j_{Tk}[h + \Delta]\right) \leq th_{+i} \tag{4}$$

where Δ may vary from $0 \ldots 2 \times \delta$ and $th_{+/-i}$ are established using the last 25 accepted records but only updated after 20 records are accepted (see Sections 4.2.2.3 and 4.3.3 of [12]).

4.2 Attacks Against PLI Approach

Type I Attack. For the type one attack the attacker attempts to replay the synchronisation portion of the original waveform, but with a different payload, using the lowest performing AWG possible.

Type II Attack. As an example of a type two attack, let us assume that the attacker is still attempting to produce a high fidelity copy of the targeted device's signal but wishes to compensate for the inherent error of their DAC so that a lower performing AWG can be used. If the error distribution of the DAC is such that it is just as likely to overshoot the desired output value as undershoot it, for the attacker to maximise the amount of allowable error between the forged signals and the authentic signals they should construct a single frame based upon the average of multiple observed waveforms and transmit it with a custom payload. The proof follows.

Following the procedure set out in Section 4.3.3 of [12], the thresholds for the next m records from device i are determined by taking the mean of distance measures for the previous n records and adding, for the upper threshold, or subtracting, for the lower threshold, the standard deviation of those same measures times some constant, K. Allowing the output of the distance measure for the j^{th} record to be represented by $d^j = d(L_{Ri}, f(s^j_i, trg^j_i))$ the thresholds are then

$$th_{+/-i}(d^j \cdots d^{j+m-1}) = \mu(d^{j-n} \cdots d^{j-1}) \pm K \times \sigma(d^{j-n} \cdots d^{j-1}) \tag{5}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ are the mean and standard deviation, respectively.

As $d(\cdot)$ is the sum of products, forging a signal that produces $(th_+ + th_-)/2$ allows for the maximum, equal amount of deviation for each sample point in either direction. The average of the signals used to calculate the thresholds is just such a signal.

We note that $d(\cdot)$ for this PLI approach is effectively using correlation to find the maximum alignment between L_R and L_T, and by extension the records, s_R and s_T, used to create the feature vectors. Allow L_{T^*} to equal those elements of

L_T found to produce the maximum output of the distance measure with L_R and l to be number of elements of L_R (i.e. we extract the $m-n$ contiguous sample points from s_T that produce the maximum correlation with the $m-n$ sample points of s_R that constitute L_R). The distance measure for the j^{th} record may then be simplified to

$$d^j = \sum_{k=1}^{l} L_R[k] \times L^j_{T^*}[k] \tag{6a}$$

$$= L_R \cdot L^j_{T^*} \tag{6b}$$

The mean of the distance measure for n training records can be expressed by

$$\mu(d^1 \cdots d^n) = \frac{d^1 + d^2 +, \ldots, + d^n}{n} \tag{7a}$$

$$= \frac{L_R \cdot L^1_{T^*} + L_R \cdot L^2_{T^*} +, \ldots, + L_R \cdot L^n_{T^*}}{n} \tag{7b}$$

$$= \frac{L_R \cdot \left(L^1_{T^*} + L^2_{T^*} +, \ldots, + L^n_{T^*}\right)}{n} \tag{7c}$$

$$= L_R \cdot \mu \left(L^1_{T^*} \cdots \text{Ł}^n_{T^*}\right) \tag{7d}$$

It is worth noting that although an infinite number of arbitrary signals (though not an infinite number of signals falling within the guidelines set by the 802.3 standard [17]) could be generated to produce a distance measure equal to the mean of the previous n records, finding the average signal only requires that an attacker observe n waveforms, align, and then average them. Of course an attacker could not know the which frames would exactly constitute the n training records, and while the attacker can align and average observed waveforms, there is no guarantee that the resulting signal, even if reproduced perfectly, would be aligned with L_R in such a way as to produce a distance measure of $(th_+ + th_-)/2$.

4.3 Experimental Validation of PLI Approach

To ensure that the devices we intended to forge were identifiable using the matched filter PLI system we collected data from 27 different Ethernet cards; using the matched filter PLI approach outlined above, we were able to identify the cards with $\approx 94\%$ accuracy (false-reject rate of 0.2%).

Our experimental setup consisted of two PCs: one to act as the Test PC (TPC), which housed the Ethernet card to be fingerprinted, while the other, the Data Acquisition PC (DAQPC), made use of a passively tapped internal Ethernet card to capture Ethernet frames sent to it over a crossover cable by the TPC. A Tektronix 4032 digital phosphor oscilloscope (DPO), interfaced via USB and controlled by MATLAB, was used as the PLI system sampler. As per our threat model, both the attacker and the PLI system used the data collected by the DAQPC.

In order to generate traffic for the DAQPC to capture, the TPC was instructed to ping the DAQPC. During a typical data acquisition period the TPC would

ping the DAQPC 10,000 times over the course of approximately three hours. To ensure that only traffic from the TPC was captured and that the measurement equipment did not affect the load characteristics of the DAQPC, as seen by the TPC, only the receiving pins of the DAQPC's Ethernet card on the secondary side of the transformer were connected to the oscilloscope. In this way the DAQPC could respond to the TPC's pings and ensure that the data acquisition process didn't cause packet loss or affect the transmitting circuitry of the TPC. Upon detection of an Ethernet frame (a simple slope-based threshold was used) the oscilloscope began to sample the signal at a rate of 2.5 Gigasamples/s; the signal was sampled 1,000,000 times, for a total of 400 micro-seconds. The oscilloscope had 8-bits of resolution.

Finally, the data collected during sampling was sent to the DAQPC via USB interface, where a MATLAB routine monitoring the interface accepted the data and stored the values in a vector called a record, which was subsequently written to disc. Each captured frame was stored in its own record; all of the records collected for a device during a session are said to encompass its dataset.

We note that a 10Mb Ethernet frame is transmitted using a differential signal to lessen the effects of environmental noise. The frame is reconstructed at the receiver by taking the difference of the received signals. In what follows, we apply the framework to the reconstructed 10Mb Ethernet waveform, which we found by taking the difference of the signals captured at the receive pins on secondary side of the DAQPC's transformer. This results in a loosening of the constraints placed on an attacker, as in actuality an attacker would be required to forge two signals when attempting to defeat the system. We make this simplification as the PLI approach of Gerdes *et al.* uses the reconstructed signal for identification.

In addition, as each channel of the oscilloscope used to acquire device signals had 8 bits of resolution, and we take the difference between the channels to reconstruct the Ethernet Frame, the device signals should actually be considered 9-bit: the maximum of the absolute value of any of the binary sample points that make up the waveforms was greater than 127 but less than 255; 8 bits, plus another bit for the sign, are required to represent this data then. The y-scale, or voltage, increment used in the capturing routine was 0.02 volts, which leads to an effective full-scale voltage of -5.12 to +5.10 V (binary values for the sample points range from -256 to 255).

4.4 Cost Function Estimate

To estimate the cost of acquiring an arbitrarily specified AWG, we assumed a linear relationship between cost and each performance parameter; i.e. we assumed that DAC performance scales linearly with cost, so that, for example, all other parameters being equal, a DAC with an SNR of 65 dB would cost more than one with an SNR of 50 dB.

Pricing information for 37 different DACs from Analog Devices was obtained using their online tool *ADIsimDAC*, which suggests DACs that meet certain user specifications, along with their cost [39]. Since we wished to obtain pricing data on as many DACs as possible, we only specified the dynamic range (-4 to 4 V) and minimum sampling resolution of 100 MS/s. We note that even though the PLI system sampler has a dynamic range of $\approx\pm 5.12$ V, only $\approx\pm 4$

V is necessary to forge the reconstructed Ethernet frame, as the signal does not exceed ±3.5 V. Furthermore, while an attacker could utilise a DAC with a different dynamic range, by scaling and applying an offset to the DAC output using an amplifier, this would introduce additional distortion and noise that would need to be included in the AWG model [40].

Having found DACs meeting these two specifications, we then extracted sample rate, resolution, noise, and distortion parameters from their datasheets. Of the 37 DACs meeting our requirements, 17 reported inter-modulation distortion (IMD) and noise-spectral density (NSD) instead of THD/SNR. While these measures could be used with our framework, by using different test signals with the distortion model and performing different noise measurements, they are nonetheless incompatible with—i.e. cannot be converted to—THD/SNR measures; as such, they were discarded. Seven DACs reported THD/SINAD instead of THD/SNR; because of the relationship between THD, SINAD, and SNR noted in Section 2.3 we were able to convert the SINAD measure to SNR. If multiple test signals or bandwidths were used to give a range of values for a particular parameter, we selected the signal with the highest frequency, at the highest output current, with measurements made over the largest bandwidth.

Using these data we performed a multiple linear regression (R^2 of 0.8185) to obtain the following cost function

$$f_c(P, Q, n, snr, thd) = 0.0693 \times P/Q \times 2500 + 1.6201 \times n - 0.1518 \times thd + 0.0164 \times snr - 26.4959 \quad (8)$$

Where the sampling rate is defined, in units of Megasamples/s, as a fraction of the PLI system sampling rate $f_p = 2500$, resolution (n) in bits, THD (thd) in dBc, and SNR (snr) in dB.

When examining the datasheets, we noticed that in general DACs with higher resolution and sample rate tended to have higher THD. This implies that is very costly to achieve small amounts of distortion at higher resolutions and sampling rates. However, when linear regression was performed using THD values from the datasheets, a positive coefficient was reported. As THD is negative, increasing the absolute value of the THD (i.e. decreasing the distortion) within the framework would actually lead to a lowering of the cost. Thus, a solver employing a cost function with a positive coefficient for THD would tend to drive it to $-\infty$ (zero distortion). To counter this we transformed the THD values by adding a positive scalar greater than any of the THD values and taking the negative of the result.

4.5 PLI System Evaluation Setup and Results

To evaluate the security of the PLI system, we first incorporated (4) into the cost-minimisation formulation given in (3), which lead to

$$\min_{f,n,snr,thd} f_c(f, n, snr, thd) \text{ subject to } d(L_R, f(s_a, trg_a)) \leq th_+ \quad th_- \leq d(L_R, f(s_a, trg_a)) \quad (9)$$

where $d(\cdot) = \max\left(\sum_{i=1}^{n} L_R[i] \times L_T[i + \Delta]\right)$, n is number of elements in L_R, $s_a = AWG(s_d, f, n, snr, thd)$, and trg_a is the sample point in the attacker's record, s_a, at which the PLI system sampler triggered.

Equation 9 is then used, along with the cost function defined by (8), to find the lowest-cost AWG necessary to successfully carry out a type one replay attack and a type two attack against each of the devices used for the experimental validation of the PLI. A randomly selected record was used in the type one attack, while the type two attack used a synthesised record based on the average of 25 records.

A lower bound for each of the AWG parameters was established by decreasing or increasing their value (the former in the case of sampling rate and resolution and the latter for SNR and THD), while the other parameters were set to their ideal values, until either of the constraints of (9) were violated by the resulting record. The lower bounds were found to be $f_a = 2/100 \times 2500 = 50$, $n = 5$, $thd = -25$, and $snr = 20$; should any one of the parameters fall below these values, the resulting record would be automatically rejected. Upper bounds were $f_a = 2500$, $n = 9$, $thd = -90$, and $snr = 100$.

Record Selection. To select the record(s) to be forged, we first chose 44 sequential records (the first record was chosen randomly, though it had to number 1000 or greater to ensure that the device was operating outside the warming-up period); the first 25 records were used to establish thresholds for the remaining 19. For the type one attack, one of the 19 records was chosen, at random, to be reproduced using the AWG model; for the type two attack a combination of 25 records were chosen from the training records and the remaining 19, with at most 24 records selected from the training set (again, these were selected sequentially). To create the averaged record, each of the 25 selected records was aligned with the first and the average computed. The reference features were extracted from the first record of each device's dataset.

In [12], 25 records are used to establish thresholds for the next 20 records. We limited our selection of records usable for forgery to only the next 19 (and stipulated that the attacker could only use at most 24 of 25 training records for averaging) because if record 20 should be selected randomly (or the attacker begins averaging with the first record), the attacker would be forging a record used as training data to determine the thresholds for the forgery. This case should be examined separately to see how much, if any, advantage is gained by the attacker. We also checked to be sure that the single record used in the type one attack would have been accepted by the PLI system—an attacker would not be able to guarantee this, which is another reason for them to use an average of several records.

Results. A summary of the AWG characteristics for each of the attacks, found using the genetic algorithm solver included in the Global Optimisation toolbox for MATLAB, are given in Tables 1a and 1b. As can be seen from examining the best-case scenario (when the attacker is required to utilise the most expensive AWG), the sampling rate and resolution of an AWG necessary to defeat a matched-filter based PLI system would need to be substantially less than those

Table 1. Characteristics of highest, mean (rounded), and lowest cost AWGs required to carry out the (a) type one attack using randomly selected signal and (b) type two attack

Parameter	Highest	Mean	Lowest
Resolution (bits)	5	5	5
Sample rate (MS/s)	53	50	50
THD (dBc)	-35	-30	-26
SNR (dB)	21	22	20

(a)

Parameter	Highest	Mean	Lowest
Resolution (bits)	5	5	5
Sample rate (MS/s)	53	51	50
THD (dBc)	-37	-32	-25
SNR (dB)	22	22	20

(b)

of the sampler used in our implementation (for the PLI system sampler, $n = 9$ and $f_p = 2500$ MS/s). In the worst-case scenario (lowest cost to attacker), the sampling rate, resolution, SNR, and THD are at the lower bounds (or nearly so), while for the mean case only the SNR and THD are appreciably distant from the lower bounds. In any case, the sampling rate, resolutions SNR, and THD of each of the DACs used for the cost function estimation of Section 4.4 are in excess of those reported in the tables.

Both the average and maximum costs for the type two attack are (slightly) higher than those of the type one, contrary to the results of Section 4.2. This is in spite of the fact that when the AWG attacker's averaged record was tested directly (i.e. it did not pass through the AWG model) with the reference feature set the resulting distance measure was almost exactly $(th_+ + th_-)/2$. It seems possible that the averaged sample point values, when they are discretised, are biased slightly towards one of the higher or lower level, instead of being equally distributed among the two (as assumed in our proof).

It should also be mentioned that because of the randomness of the noise an attacker record will sometimes be rejected at the reported minimum SNRs. Having repeatedly checked for constraint violations using the same SNR, it appears that the more the noise changes the trigger point of the attacker record relative to the record used for the reference feature set (i.e. as $|trg_p - trg_a|$ grows larger) the more likely it is that the record will be identified as a forgery. To ensure acceptance, the attacker should employ an AWG with a slightly higher SNR ($\approx$2 dB).

5 Conclusion

We have proposed, and illustrated the use of, a framework to determine whether an attacker could defeat a given PLI system by replaying a record using an AWG of a specified sample rate, resolution, THD, and SNR. The framework is flexible enough to be used in evaluating arbitrary PLI system implementations, using different threat models and AWG models. We also showed how the framework can be used with a cost-minimisation problem to find the lowest performing AWG necessary to defeat a PLI system. Given a particular pricing model for the sample rate, resolution, THD, and SNR, the cost-minimisation formulation can also be used to determine the most cost-effective AWG.

For the reasons given in Section 2.2, this version of the framework did not incorporate channel effects and assumed ideal/identical samplers for the attacker

and PLI system. In order to better evaluate the security of PLI systems, we will extend our work by integrating both channel models and models for non-ideal/differing samplers into the framework. To widen the application of the framework, we will use it to evaluate and compare PLI approaches for the wireless domain and investigate the feature replay attack mentioned in Section 2.1. Finally, the immediate focus of our future work will be to experimentally confirm the predictions of the framework for the PLI system of Gerdes *et al.*

References

1. Gerdes, R.M., Daniels, T.E., Mina, M., Russell, S.F.: Device identification via analog signal fingerprinting: A matched filter approach. In: Proceedings of the 2006 Network and Distributed System Security Symposium (NDSS 2006). The Internet Society (2006)
2. Hall, J., Barbeau, M., Kranakis, E.: Detection of transient in radio frequency fingerprinting using signal phase. In: Proceedings of the 3rd IASTED International Conference on Wireless and Optical Communications (WOC 2003), pp. 13–18. ACTA Press (2003)
3. Hall, J., Barbeau, M., Kranakis, E.: Enhancing intrusion detection in wireless networks using radio frequency fingerprinting. In: Proceedings of Communications, Internet and Information Technology (CIIT 2004). ACTA Press (2004)
4. Ureten, O., Serinken, N.: Wireless security through rf fingerprinting. Canadian Journal of Electrical and Computer Engineering 32, 27–33 (2007)
5. Brik, V., Banerjee, S., Gruteser, M., Oh, S.: Wireless device identification with radiometric signatures. In: Proceedings of the 14th ACM International Conference on Mobile Computing and Networking (MobiCom 2008), pp. 116–127. ACM (2008)
6. Shi, Y., Jensen, M.: Improved radiometric identification of wireless devices using mimo transmission. IEEE Transactions on Information Forensics and Security 6(4), 1346–1354 (2011)
7. Rasmussen, K.B., Capkun, S.: Implications of radio fingerprinting on the security of sensor networks. In: Proceedings of the Third International Conference on Security and Privacy in Communications Networks and the Workshops (SecureComm 2007), pp. 331–340. IEEE Computer Society (2007)
8. Danev, B., Capkun, S.: Transient-based identification of wireless sensor nodes. In: Proceedings of the 2009 International Conference on Information Processing in Sensor Networks (IPSN 2009), pp. 25–36. IEEE Computer Society (2009)
9. Saparkhojayev, N., Thompson, D.R.: Matching electronic fingerprints of rfid tags using the hotelling's algorithm. In: Proceedings of the IEEE Sensors Applications Symposium (SAS), pp. 19–24. IEEE Computer Society (2009)
10. Danev, B., Heydt-Benjamin, T.S., Capkun, S.: Physical-layer identification of rfid devices. In: Proceedings of the USENIX Security Symposium (USENIX-SS 2009), pp. 199–214. USENIX Association (2009)
11. Zanetti, D., Danev, B., Capkun, S.: Physical-layer identification of uhf rfid tags. In: Proceedings of the 16th ACM Annual International Conference on Mobile Computing and Networking (MOBICOM 2010), pp. 353–364. ACM (2010)
12. Gerdes, R.M.: Physical layer identification: methodology, security, and origin of variation. PhD thesis, Iowa State University, Ames, IA (2011)
13. Danev, B.: Physical-layer Identification of Wireless Devices. PhD thesis, ETH Zurich, Zurich, Switzerland (2011)
14. Bolle, R.M., Connell, J.H., Pankanti, S., Ratha, N.K., Senior, A.W.: Guide to Biometrics. Springer (2004)

15. Danev, B., Luecken, H., Capkun, S., Defrawy, K.E.: Attacks on physical-layer identification. In: Proceedings of the Third ACM Conference on Wireless Network Security (WiSec 2010), pp. 89–98. ACM, New York (2010)
16. Edman, M., Yener, B.: Active attacks against modulation-based radiometric identification. Technical report, Rensselaer Polytechnic Institute, Department of Computer Science (2009)
17. IEEE: IEEE 802.3-2008 ieee standard for information technology-specific requirements–part 3: Carrier sense multiple access with collision detection (cmsa/cd) access method and physical layer specifications. Technical report, IEEE, IEEE Std 802.3–2008 (2008)
18. Burns, M., Roberts, G.W.: An introduction to mixed-signal IC test and measurement. Oxford University Press (2001)
19. Kester, W.: Evaluating high speed dac performance. Technical report, Analog Devices, MT-013 Tutorial (2008)
20. Balestrieri, E., Moisa, S., Rapuano, S.: Dac static parameter specifications some critical notes. In: Proceedings of the 10th IMEKO TC-4 Workshop on ADC Modelling and Testing, vol. 1, pp. 81–86 (2005)
21. Hendriks, P.: Specifying communications dacs. IEEE Spectrum 34, 58–69 (1997)
22. Tektronix USA: AWG7000B Series AWG Data Sheet
23. Oppenheim, A.V., Schafer, R.W. (eds.): Discrete-Time Signal Processing. Prentice Hall (1989)
24. Analog Devices USA: (AD9734/AD9735/AD9736 Series DAC Data Sheet)
25. Wambacq, P., Sansen, W.M. (eds.): Distortion Analysis of Analog Integrated Circuits. Springer (1998)
26. Andersson, K.O.: Studies on Performance Limitations in CMOS DACs. PhD thesis, Linkopings universitet, Linkoping, Sweden (2002)
27. Wikner, J.J.: Studies on CMOS Digital-to-Analog Converters. PhD thesis, Linkopings universitet, Linkoping, Sweden (2001)
28. Chan, K.L., Zhu, J., Galton, I.: Dynamic element matching to prevent nonlinear distortion from pulse-shape mismatches in high-resolution dacs. IEEE Journal of Solid-State Circuits 43(9), 2067–2078 (2008)
29. Naoues, M., Morche, D., Dehos, C., Barrak, R., Ghazes, A.: Novel behavioral dac modeling technique for wirelesshd system specification. In: Proceedings of the IEEE Electronics, Circuits and Systems (ICECS 2009), pp. 543–546 (2009)
30. Wit, P.D., Gielen, G.: Efficient simulation model for dac dynamic properties. In: Proceedings of the IEEE Circuits and Systems (ISCAS 2010), pp. 2896–2899 (2010)
31. Andersson, N.U., Andersson, K.O., Vesterbacka, M., Wikner, J.J.: Models and implementation of a dynamic element matching dac. Analog Integrated Circuits and Signal Processing 34, 7–16 (2003)
32. Vandenbussche, J., der Plas, G.V., Gielen, G., Sansen, W.: Behavioral model of reusable d/a converters. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 46, 1323–1326 (1999)
33. Riley, K., Hummels, D., Irons, F., Rundell, A.: Dynamic compensation of digital to analog converters. In: Proceedings of the IEEE Instrumentation and Measurement Technology Conference (IMTC 1999), vol. 2, pp. 1310–1315 (1999)
34. Gerdes, R.M., Mina, M., Daniels, T.E.: Awg characterisation definitions. Technical report (2012), `http://www.eng.usu.edu/ece/faculty/rgerdes/papers/tech/awgCharDef.pdf`
35. Analog Devices USA: AD9763/AD9765/AD9767 Series DACData Sheet
36. Maloberti, F., Estrada, P., Valero, A., Malcovati, P.: Behavioral modeling and simulation of data converters. In: Proceedings of IMEKO 2000, vol. 10, pp. 229–236 (2000)

37. Liu, E.W.Y.: Analog Behavioral Simulation and Modeling. PhD thesis, University of California, Berkeley, CA (1993)
38. Analog Devices USA: AD9777 Series Data DAC Sheet
39. Analog Devices: Design Tools ADIsimDAC (2011), http://designtools.analog.com/dtSimDACWeb/dtSimDACMain.aspx
40. Aksin, D.Y., Maloberti, F.: Non-linear behavioral model of a bipolar track and hold amplifier for high-speed and high-resolution adcs. In: Proceedings of the IEEE Electronics, Circuits and Systems (ICECS 2005), pp. 1–4 (2005)