# Fillory of PHY: Toward a Periodic Table of Signal Corruption Exploits and Polyglots in Digital Radio

*Sergey Bratus, Travis Goodspeed, Ange Albertini, Debanjum S. Solanky*

## Abstract

Boundaries between layers of digital radio protocols have been breached by techniques like packet-in-packet: an attacker controlling the application layer payloads can, in fact, inject frames into lower layers such as PHY and LNK. But can a digital transmitter designed for a particular PHY inject frames into a different, *non-compatible* PHY network?

We present several case studies of such *cross-protocol injection*, and show that non-compatible radio PHYs sharing the same frequencies need not merely collide and jam each other, but can instead unexpectedly cross-talk. We propose a methodology for discovering such cross-talking PHYs systematically rather that serendipitously. No PHY is an island.

## 1 Introduction

**Motivation.** A key method of offensive research is to explore the behavior of a system when its input or state are corrupted. What states—legal or illegal, expected or unexpected—can be reached then? Can some components of the system be made to disagree about the system's state or the nature of the inputs? Is there a chain of corruptions that can bring the system to the state desired by the attacker and/or unexpected by the designers?

Closely related to these questions is the idea of the degrees of freedom by which corruption may occur. For example, the attacker may only be able to manipulate data payloads of a protocol or input format, and not its metadata—or vice versa. The corruption may be precisely controlled by the attacker, or be probabilistic and rely on random ambient noise—as in the packet-in-packet example discussed below—or artificially created noise, such as is caused by heat and radiation. The attacker may be able to inject arbitrarily crafted corrupted messages at a particular level—such as raw protocol frames, bit-by-bit—or may only be able to corrupt only specific parts of messages sent by others, and so on. Engineering of attacks and mitigations depends on systematic exploration of these degrees of freedom.

This space for memory corruptions in programs is well-studied. Injection of corrupted states via crafted inputs arguably comes first [12, 15], followed by random corruption caused by external physical interference and—more recently—by cross-talk [14, 10]. The effects of causing components of a system (distributed or monolithic) to disagree in the interpretation of messages are less explored, but their power has been demonstrated, e.g., by the "PKI Layer Cake" [9] and "Android Master Key" vulnerabilities.[1] Input polyglots—files that are interpreted by different programs as containing data in their "native" formats, such as valid PDF and ZIP at the same time—also received considerable attention.[1]

At the same time, in wireless systems such questions have largely been overlooked for corruptions other than jamming and collisions; for the latter, sophisticated models have been developed (e.g., [16, 13]). To fill this gap, in 2010 we started a systematic exploration of the lower layers of digital radio protocols that led us to attacks such as *packet-in-packet* for 802.15.4 [6], *active receiver fingerprinting* techniques for 802.11 and 802.15.4, and *local dialect/shaped charges* attacks [2, 8].

In particular, we demonstrated that a remote attacker can use control of the application payload of 802.15.4 frames to inject a raw PHY frame into the link—without a radio, by leveraging ambient noise. We further demonstrated that frames can be manipulated to be selectively received by some radio chips but not others, regardless of signal strength or signal-to-noise ratio. We showed that a receiver's view of valid received frame may share no symbols with the actual transmitter's view, which inval-

*Fillory is an imaginary land of cross-communicating species (talking animals). Fillory turns out to be real.

[1] See `http://www.saurik.com/id/17`, `http://www.saurik.com/id/18`, `http://www.saurik.com/id/19`

idates a class of defenses based on "escaping" symbols such as the start-of-frame-delimiter occurring in the payload of a frame. [5]

We summarize these previous findings in Section 2.1.

We continue this exploration in this paper, this time focusing on the corruptions, misconceptions, and degrees of freedom in the digital radio PHY layer exclusively. Specifically, we ask the following questions:

- Can a digital radio inject a message into a PHY that it was not designed to inter-operate with, so that the injected message is received as valid by the standard radios of other, incompatible PHYs?
- Can a single PHY message be received by standard radios of incompatible PHYs, and be interpreted by all of them as valid messages with *different* non-trivial content?

We demonstrate that at least for some popular PHYs using common modulation schemes the answers are "yes" and "yes". We then discuss the degrees of freedom in creating such signals—i.e., what parts of the PHY protocol the attacker can manipulate/corrupt and why.

In our exploration, we use shortwave digital radio protocols used for texting in amateur radio. Our choice is directed by their relative simplicity and ease of constructing their physical signals: the signal processing tasks can be handled by PC sound cards after the signal is downshifted from the short wave range frequencies. Despite their simplicity, these protocols use the same standard modulation schemes such as frequency, phrase, and amplitude shift keying as the more complex protocols. Thus our techniques are not inherently limited to short wave or to specific protocols.

**A toy example.** Imagine a simple digital radio built for sending short text messages, over large distances and noise, at just about the speed with which humans can type them. Such protocols are easy to understand because they tend to use the simplest ways of modulating the signal and encoding the messages. We are more interested in modulation, as it lies deeper in the PHY than the encoding, and determines the basic design of the receiving circuits—whereas the encoding sublayer of PHY, generally speaking, interprets what the receiving circuit feeds it.

Each PHY, depending on its modulation, requires a different analog circuit that must handle the physical signal before it can be conveniently passed to digital ones for decoding. These analog circuits are built to do their job at the least cost and with reasonable resistance to the irrelevant aspects of the signal, which may be noise or sender artifacts. Thus we think of PHYs using different modulations as—at worst—interfering with each other when trying to use the same frequencies, but never intelligibly cross-talking. But is this correct?

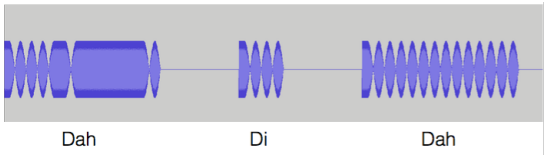


Figure 1: This PSK31 signal is also Morse code for 'K'

Consider, for example, a radio that switches the phase of the carrier to encode 1s and 0s (we will discuss an example of such a digital protocol, PSK31, used broadly in amateur radio texting). Its receiver is built to ignore amplitude changes, and focuses only on the phase changes in the incoming wave, no matter how weak or strong the signal is.

At the same time, consider a simple amplitude-modulated Morse code signal, such as the series of peeps you occasionally hear on the radio. Some peeps are shorter, others are longer; to approximate the way they sound to the human ear, they are vocalized as *di*s and *dah*s (A is thus di-dah, B is dah-di-di-di-dit, C dah-di-dah-dit, etc.) This is, of course, a different PHY, used both in radio and other wavelengths (e.g., IrDA uses a similar modulation).

It turns out that what you are hearing in valid and sensible Morse code may—at the same time—be an entirely different message encoded in PSK31. Thus the two messages occupy the same frequency at the same time, and will be received as completely different by two different and *standard* receivers, neither suspecting the existence of the other, and both believing that what they received owned the channel loud and clear!

More importantly, the PSK31 radio is taking to a *different PHY*—built for both the different modulation and encoding!

You can see an example of this cross-PHY signal in Fig. 1. Since phase changes are normally carried out at zero amplitude to avoid artifacts, you can actually see where the signal's phase switches; these switches give the Morse Dis and Dahs a little wobble—but it hardly interferes with the reception. Note that the two Morse Dahs are quite different, and carry different PSK31 digital payloads; on the other hand, a PSK31 receiver does not care about changes in amplitude, only phase!

**Isn't this like steganography?** Yes and no. In short, the problems of PHY cross-talk and steganography differ in both their constraints and their threat models. Steganography exists in a very different design space, where the representation of the hidden message can be

arbitrary, and—at best—recognizable with only a specialized receiver. In cross-PHY, by contrast, we target well-known existing PHY receivers. Steganography presumes the model of safely communicating past a listening adversary; in the PHY cross-talk model, our concern is rather with reaching targeted radios via a non-compatible radio in the target's "radio neighborhood" that we happen to control.

In this paper, we pose the problem of cross-talking PHYs, give several simple but counter-intuitive examples, and offer the framework for thinking about the problem—and finding more of it systematically.

**How to read this paper.** There are two ways to read this paper. If you prefer seeing the PoC first, skip to Section 4; if, on the other hand, you prefer a discussion of what it means, what common misconceptions it explodes, and where it fits in other attacks on PHY, continue reading the next Section 2. Section 3 reviews a few basic concepts of radio design, so you may skip it on your way to the PoC if you have a background in radio. Finally, Section 5 joins these two threads and calls for more research.

## 2 The Figments of PHY

**Foundations: Injection and Monitoring** Practical network security starts with two basic questions: what the attacker needs to inject lower-layer (PHY or LNK) frames into the network, and how the defender can monitor the entirety of traffic at these layers. Advances in affordable tools for these two tasks change the playing field.[2]

For digital radio networks, the answers harbor many surprises. For instance, consider these questions:

1. Can a digital radio receive a valid, non-corrupted frame that was never transmitted *as such* by any radio of the same PHY?

2. Can a digital radio receive a valid, non-corrupted frame that shares *no bytes* with any frame transmitted by any radio of the same PHY?

The answers for these questions are "yes" and "yes". We survey the respective prior work below. However, this prior work has one intuitive but—as it turns out—strong limitation: it only considers frames coming from the radios *built for the same PHY*.

[2]See, e.g., discussion in [2, 7]. The respective capabilities are colloquially knows as raw injection and sniffing. Advances in affordable tools granting these capabilities invariably facilitated advances in the exploration of the respective protocols. Joshua Wright observed that *"Security does not get better until tools for practical attack surface exploration are made available."* This observation is known as *Wright's Principle*.

In this paper, we remove this limitation and deal with the following questions:

1. Could a frame received as valid by a digital radio in fact come from a *non-compatible* digital radio, not intended by design to speak the receiving radio's protocol?

2. Could a frame received as valid by one digital radio (say, while it is monitoring the channel) appear, at the same type, as a valid frame to a radio built for a different protocol? Or, as it would appear, can two frames reliably occupy the same physical medium at the same time, according to two different receivers?

3. If so, can such cases be systematically discovered?

The intuitive answer is that non-compatible PHYs are non-compatible, and their sharing a medium would lead to frequent but ultimately harmless collisions—i.e., at worst a DoS or a QoS reduction, not a potential vector for injecting exploit payloads.

Yet the actual answers, as we show in this paper, are also "yes", "yes", and, moreover, "very likely". These answers followed immediately from a certain way of organizing our thinking about the PHY protocols—which we demonstrate, in hopes of it being adopted and developed.

### 2.1 Prior art: Boundaries expected, but not enforced

In a nutshell, many boundaries that intuitively hold for other kinds of networks—and are assumed by protocol designers to similarly hold in digital radio—turn out to be purely imaginary, and unexpectedly permeable to attackers. Intuitions that guided the design of the "wired" Internet protocol stack models—e.g., that frames are either received exactly as they were sent or are corrupted by noise and can be easily recognized as such and discarded—fail, and fail most unexpectedly.

**Cross-layer injection** Goodspeed et al showed that injecting frames into an unencrypted 802.15.4 network can be achieved by a remote attacker with as little as controlling the *application* layer payloads of the protocol—without ever owning an 802.15.4 radio! [6] This means that a received apparently valid frame *may have never been knowingly sent as such* by any compatible radio. Moreover, as further work [5] showed, the received valid frame need not even *share any bytes with any frame* that was transmitted—and so escaping certain nybbles in transmitted frames to avoid the radio unwittingly enabling the above *packet-in-packet attack* would not work.

**Chipset-specific reception** Not only sending, but also receiving PHY frames in digital radio harbors surprises. For example, [2] and [8] showed that the same crafted frame could be reliably seen by some 802.11/Wi-Fi and 802.15.4/ZigBee radio chips as valid, while being seen as invalid or indistinguishable from noise by others—regardless of its signal strength and actual ambient noise. Moreover, such ways of frame-crafting were rich enough to fingerprint the receiving radio's chipsets. Thus, e.g., simple manipulations of the frame's preamble could render an exploit frame invisible to a WIDS while it successfully reaches its target that uses a different chipset.

In each of the above cases, attacker success is predicated on the protocol designers having made certain most basic and intuitive assumptions about the natural separation of layers—both OSI model-granular and at the internal sub-layers typical of a PHY implementation, but still distinct for all engineering purposes. This separation proved imaginary—and so did the value of security models based on it.

With such basic and intuitive assumptions shown to be false for digital radios, we ask: what other assumptions may be false, and how can we go about enumerating them *systematically* rather than serendipitously?

## 2.2 Our contributions

**Cross-PHY injection** In this paper, we show several case studies of digital radios designed for a specific PHY layer successfully injecting signals into another.

**A systematic approach** We show that these case studies, rather than from being discovered serendipitously, come from a way of enumerating the design features of the different PHYs, arranging them into explicit families by similarity of these features—and then examining the effects of noise, scrambling and whitening (if any), encryption (if any), etc. on both the protocols and known attacks. This approach will also clarify the role of features that add to the protocol complexity—do they help or hinder the attacker, and which ones are actually more helpful than others, by design or pure serendipity?

## 2.3 Why this matters

A decade ago these questions might have been thought purely theoretical, as operating a digital radio network (even Wi-Fi) represented a deliberate investment. These days, as our environment gets saturated with wireless remotely accessible devices that contain several kinds of digital radios—a smartphone can have up to four or five besides its baseband connection—having a compromised RF-capable device near one's network is no longer theoretical.

A compromised, attacker-controlled device may not have a radio chip for a compatible protocol—but can it nevertheless cross-talk to your network? This possibility is no longer trivial to dismiss. Moreover, it is reasonable to expect that designers of different protocols considered accidental RF interference but not malicious cross-talk.

The output of network monitors is another concern. When a monitor captures a PHY frame, the usual operator assumption is that the signal contains that frame and *nothing else*—but with "polyglot" signals, appearing as different frames to different receivers, this is no longer a safe assumption.

Such polyglots have been shown to exist across many application formats [1]. This publication made a point of being distributed in PDF files that also appeared as valid ZIP files, which were also valid PNG images, and, to boot, valid bootable OS volumes, TrueCrypt containers for a particular key, and several other formats, both plain text and encrypted.

So why not in digital radio? Indeed, we show examples of such polyglot PHYs below.

Still, our primary contribution is not the shock value of non-intuitive behaviors of a few case study protocols. Rather, it is the concept of systemic protocol relationships that lead to cross-talk. Just like a periodic table of elements made the case that chemical properties of basic substances were not random, so, we argue, are the security properties of basic PHY designs. These properties should be kept in mind when designing future protocols; no protocol is any longer an island in a world saturated with remotely reachable radios.

# 3 The Fundamentals of PHY: Marconi vs Machiavelli

This section provides a very brief overview of the building blocks of a digital radio protocol—from the point of view of corrupting and manipulating them to create cross-PHY and polyglot signals.

These building blocks tend to translate to sublayers in the implementation of that PHY, often designed by separate engineering teams. Even though the classic OSI model lumps all of PHY into a single monolithic layer, the engineering reality cannot be more different—and thus presents the attacker with a variety of nearly independent targets to manipulate rather than just one.

As always, the engineering of attacks is the dual to that of targets—and, as always, demonstrates overlooked principles of the original system. We formulate these principles here—as Machiavelli (attacker) rather than Marconi (radio engineer) might phrase them.

Before formulating these principles, however, we briefly review the basics of radio. Please feel free to skip

to 3.2 if you don't need this recap!

## 3.1 Modulation basics

Digital radio modulation schemes modify the transmitter's carrier wave to send information; the modifications are such that compatible—but far from perfectly matching or precise—receivers could tune in and extract it from the changes they observe. The natural parameters to vary about a sine carrier wave—which has the shape of the sine, $A\sin(\omega t + \theta)$—are the amplitude $A$, the frequency $\omega$, and the phase $\theta$.

These information-carrying variations become functions of time $t$, i.e., $A(t)$, $\omega(t)$, and $\theta(t)$. The receiver, generally speaking, is a circuit built for measuring values derived from one or more of these functions.

Of course, some variations are not physically orthogonal—e.g., rapid changes in amplitude at high power levels will produce noise components in frequency and phase-based demodulators. For this reason, amplitude in these modulation schemes is also varied, to reduce such effects (as we will see in 4.2).

In reality, both the timing for these measurements and the idea of the shared carrier frequency between a transmitter and a receiver are only approximate, as is their circuits' accuracy in representing the signal; noise may additionally factor into these limitations. Thus, measuring *absolute* values about a signal is rarely done; instead, it's the *relative* values that are measured, or, rather, their discrete changes or *shifts*.

Specifically, the more popular digital radio protocols are based on either Frequency Shift Keying (FSK) or Phase Shift Keying (PSK), which encode the information in discrete, step-wise changes of frequency or phase.

More generally, digital radio receivers are built for certain modulation and encoding.[3] They are designed to best resist noise and the likely kinds of interference—but hardly figure on deliberate manipulations by other PHYs (or the same PHY). This general design trend gives rise to a number of the following attack principles.

## 3.2 Attacker principles of digital radio

**1. Non-data-bearing signal parameters are deliberately ignored.** Receiver de-modulation circuits are built to measure the modulated parameter(s), and to ignore the irrelevant ones. *These deliberately ignored parameters serve as natural degrees of freedom for signal manipulation and corruption.*

[3] As well as error correction, which we do not consider in this paper, but which has also been successfully manipulated for targeted corruption schemes. Error correction in a form of rewriting, and rewriting systems provide powerful computation models—some are even Turing-complete!

Importantly, in both FSK and PSK the amplitude of the signal or its changes are purposefully ignored, inasmuch as they do not affect the processing of the signal in the receiving circuit; in fact, these changes—even though they can encode information for other PHY schemes—are by design rendered *invisible* to the receiver.

Moreover, not reacting to any variations of the signal orthogonal to the chosen modulation scheme, however strong these may be, is an advantage of these circuits from the Marconi view.

We will apply this principle to the construction of our cross-PHY polyglots.

**2. Additional granularity of signal-bearing changes is deliberately ignored.** A receiving circuit built for a fixed number of discrete values of the parameter it measures will ignore further variations of that parameter. This provides an additional degree of freedom for the attacker.

In particular, the shifts in frequency or phase in the actual signal need not exactly match those expected by the demodulating circuit. For example, a particular PHY modulation may target the shifts between two relative frequencies or phases, as in 2FSK or 2PSK—but the actual signal may use four or more of these (e.g., 4FSK or 4PSK), attenuated by the amplitude changes to minimize the conversion effects.

Thus the actual signal may encode more information than the targeted receiver circuit, optimized for noisier environments, may be able to extract—but other circuits would happily extract it, from the same signal.

**3. Background noise is not arbitrary, and can be leveraged.** Although noise is typically modeled as being random, it is often not so. The interpretation of noise by a receiver may be strongly biased toward certain symbol values, by demodulation, encoding, error correction, or a combination thereof. These biases represent another degree of freedom for the attacker, if only probabilistically realized.

For example, 2FSK noise represented as bytes may be biased toward values of 0x00, 0xFF, 0xAA, 0x55, as was the case in the NordicRF scheme described in [4]. The latter allowed sniffing of packets with unknown MAC addresses, despite the scheme being deliberately designed to make this hard.

As another example, the Packet-in-packet attack [6] is enabled by the typical 802.15.4 noise being likely to corrupt only a few symbols in a packet, rather than larger swathes of symbols. Thus the probability of the enclosing frame's SFD being corrupted by noise without any of the following payload of the enclosed crafted "frame" in the payload being corrupted at the same time is empirically high enough to allow the attack to succeed.

**4. Layer and sublayer boundaries can be breached with the right degrees of freedom.** A combination of the above principles may work across protocol layers. With the right degrees of freedom, controlling only application data can emulate lower layers of another protocol's PHY, or produce corrupted versions thereof that would never be sent by a compliant compatible PHY.

For example, the Packet-in-packet attack allows injection into raw PHY to attackers controlling only 802.15.4 application data—due to self-similarity of the protocol. It can be said that packet-in-packet establishes a polyglot between the protocol's payload and signaling, realized by the presence of noise.

# 4 The Specimens of Cross-PHY Injection and Polyglots

In this section, we give the simplest examples of cross-talking digital radio PHYs, based on the popular Amateur Radio texting protocols that we started exploring in the introduction.

**Simple, but important.** Before we dive into these simpler examples, though, we should stress that their simplicity should not belie their importance. Their building blocks such as PSK modulation are still used by much more complex protocols such as 802.15.4 and even 802.11.

Raw signal crafting and sniffing of these more complex protocols can no longer be accomplished by means of a PC sound card, as is the case of RTTY and PSK31.[4] Still, construction of custom radios for such injection is getting cheaper as software-defined radio platforms progress (cf. the evolution of SDRs from the costly USRP by Ettus Research to the affordable Jawbreaker and HackRF platforms by Great Scott Gadgets.[5])

Complexity of a radio PHY offers no better security than other kinds of obscurity. There was a time when the complexity of Wi-Fi cards was thought to preclude raw injection, making Wi-Fi opaque and "secure". Then firmware hacks of the Prism and Atheros chipsets made it possible—and opened the floodgates of ring-0 driver bugs.

## 4.1 OOK, FSK, PSK

The simplest modulation scheme of this kind is On-Off Keying, a.k.a. OOK[6] In OOK, a high amplitude of the carrier wave means "1", a lower (or absent) amplitude means "0". The Morse code is a common choice for encoding text messages on top of OOK; other PHYs such as IrDA use OOK it with different encodings.

However, OOK is more sensitive to noise when implemented in a cost-efficient manner. That is why it has been supplanted first by Frequency Shift Keying (FSK) in the RTTY radioteletype protocol, then by Phase Shift Keying (PSK) in RTTY's successor PSK31. In RTTY, the sender switches the power between two frequencies (making it a 2FSK modulation), in PSK31 between two phases (making it 2PSK, a.k.a. BPSK, or, since the two phases are inverse to each other, phase-reversal keying). RTTY started its service in military use in the 1930–40s, and passed into amateur use in 1970s, where it was later replaced by the more efficient PSK31; PSK31 remains popular to this day.

## 4.2 PSK31 / OOK polyglots

PSK31, in its commonly used non-error-correcting mode, employs a BPSK modulation, which switches between the two opposite ($180^{\circ}$-separated) phases of the carrier. A phase shift means a "0", no phase shift means "1". The receiver samples the shifts at the rate of 31.25 Baud, which, together with the modulation, accounts for the protocol's name, and was chosen to make signal processing with 8-bit PC sound cards easy.[7]

In order to reduce the artifacts of the shifts in the frequency domain—when made as full amplitude, they'd produce frightful boundary effects (see Fig. 2)—an amplitude envelope is used, so that the shifts could be done at zero amplitude. As it stands, the protocol needs about $60Hz$ of bandwidth, and, with this narrow bandwidth, allows multiple simultaneous conversations worldwide over short wave radio, atmospheric conditions permitting.

Although under this envelope the amplitude changes with the signal, its changes are not intended to carry any information, and are not measured for such by the receiver. Only the phase matters.

This indifference to amplitude is precisely what makes PSK an excellent cross-PHY talking animal. Figure 3 shows the actual waterfall display of the signal we previously depicted in Figure 1. (Note the artifacts from the boundary effects brought on by OOK.) By the way, the PSK31 payload of this polyglot signal also encodes 'K,' in its first Dah group, whereas both the Di and the second Dah are all zeros.

[4] E.g., by means of the excellent and free Fldigi program suite, `www.w1hkj.com/`

[5] See, e.g., `https://greatscottgadgets.com/hackrf/`

[6] This *may* be the protocol that The Librarian in Sir Terry Pratchett's "Discworld" series has been using to communicate; unfortunately, since Amateur Radio was not developed in the known parts of Discworld, his attempts reportedly got no farther than the handshake specifying the protocol. Cf. RFC 4253.

[7] http://www.arrl.org/psk31-spec

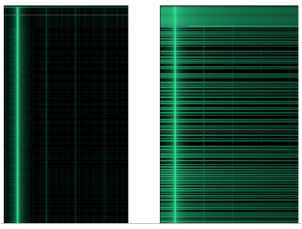

Figure 2: PSK31 in the frequency domain ("waterfall" display) with and without the amplitude envelope

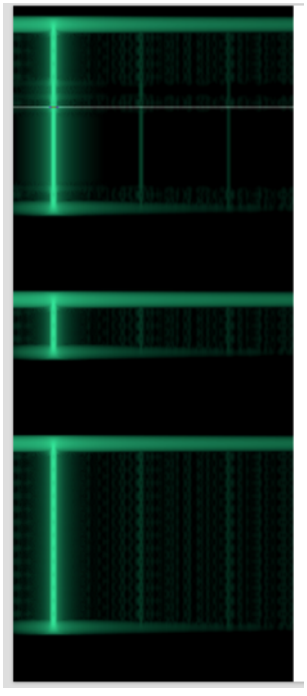

Figure 3: PSK31 / OOK polyglot in a "waterfall" display

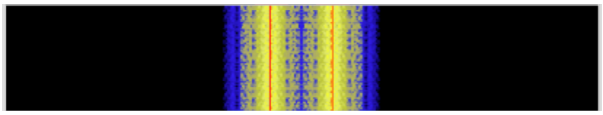

Figure 4: An RTTY 2FSK transmission in a "waterfall" display

**A stego-bonus: Varicode tricks & envelope ambiguity** In PSK31, both its encoding *and* its modulation scheme provide broad opportunities for classic steganographic tricks. We briefly point them out here for their niftiness; our focus is not on steganography but rather on the basics of PHYs.

PSK31 uses the *Varicode* encoding, which is more efficient for English than ASCII; most common letters are short. In Varicode, two or more consecutive zeros serve a letter boundary marker. No character code contains more than one zero at a time, and every letter begins and ends with a one, making it convenient to detect their boundaries.

PSK31 stations use zeros to indicate idle time, which is handy for human operators who type with varying speed. Moreover, illegally long letters are ignored, which allows for extensions, such as addition of non-English alphabets for agreeing stations, without messing up the decoding in all others. Between these two encoding features, rich variations of encoding additional information are possible.

A second observation concerns the ambiguity of the PSK31 envelope and is due to Craig Heffner. Normally, PSK31 stations do not drop the amplitude when transmitting consecutive 1s (no phase shift, thus no drop needed)—but could do so, and most receivers would not notice the difference. This opens additional possibilities for manipulating the PHY without changing the legitimate PSK31 payload.

## 4.3 RTTY / PSK31 polyglots

In its simplest variant, RTTY uses 2FSK modulation, switching between a pair of frequencies, to encode 5-bit characters of the Baudot code, with 2 stop bits and no parity bit. Variations of these modulation and encoding schemes exist, including the use of both amplitude and frequency shifting; our discussion can be modified to cover these as well as the basic case.

The energy distribution in 2FSK RTTY is as shown in Figure 4. At any sampling time, the receiver compares the relative power in the pair of frequencies used, and interprets it as a shift or no-shift. The phase (barring any artifacts) is ignored; only the relative power in the frequency pair matters.

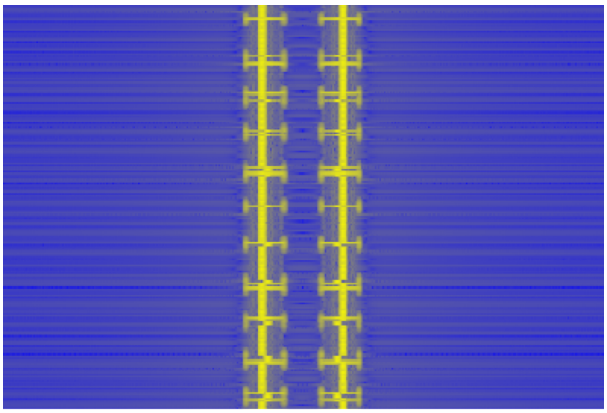

Figure 5: A PSK31/RTTY polyglot transmission in GNU Radio waterfall display.

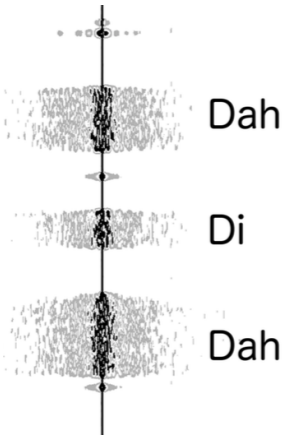


Figure 6: Ethernet frames as OOK, on a faulty CAT5 cable

PSK31, on the other hand, is tolerant to power changes. This opens the way to a RTTY/PSK31 polyglot, which uses frequency shifts to encode an RTTY message, while using phase shifts in a higher or lower frequency to encode a PSK31 one!

To construct a RTTY/PSK31 polyglot, we take two PSK31 signals far enough apart that their bands do not overlap, and modulate their relative power according to the RTTY encoding, *as if these were the two carrier frequencies of RTTY's 2FSK signal.*

We constructed PSK31/RTTY polyglots using the Python code utility GoodPSK[8] and GNU Radio code[9]. The GNU Radio waterfall display image of a polyglot is presented in Figure 5.

In this figure, several key features of the respective protocols are easy to see. The two PSK31 signals, pretending to be the two frequencies of RTTY, form two narrow bands (compare with Figure 4). These two bands are slightly wider that the original PSK31 signals due to amplitude variation artifacts; other artifacts are also visible.

We verified the successful decoding of these signals with Fldigi.

## 4.4 Madeline: an accidental Ethernet / OOK polyglot

This PoC happened accidentally, when at one point we remotely connected to the first author's PC running the software-defined short wave radio for the polyglot experiments.

The radio was picking up an unexpected signal that looked like an OOK signal—but only when a VNC connection was active. As soon as the VNC connection stopped updating the screen, so did the signal. What was going on?

It turned out to be due to a poorly crimped Ethernet cable. This cable happened to connect the PC to the switch—and VNC packets would create a signal powerful enough for the radio to pick up. Of course, is a world of TEMPEST and stealing cryptographic keys via radio emissions (e.g., [11, 3] this should be not surprising at all—but it still was, considering that the signal was detected by a commodity radio without any special tuning, with regular TCP packets.

Together, that cable, the Ethernet card, and the Linux kernel's TCP/IP stack made an effective shortwave OOK transmitter *controlled by the remote TCP endpoint*. After manipulating the TCP packet (and therefore the Ethernet frame) lengths, we obtained signals such as in Figure 6, an accidental Ethernet/OOK polyglot, with Ethernet as a digital radio transmitter.

The implications of this are not new—e.g., [11] discussed the design of Trojan software that would exfiltrate data via a Frequency Shift Keying scheme by controlling a CRT monitor and causing particular shortwave patterns of emissions—but they are nevertheless worth stressing. After all, our polyglot required no local Trojan and no elaborate radio setup—just a faulty cable, which otherwise served well and without notice of its nefarious flaws.

Controlling the size of Ethernet frames crossing to a PC is easy enough for a server, no matter what TCP client program it is interacting with—the client, likely, does not in fact have much of a say in the matter, due to the nature of IP and Ethernet, which wraps incoming TCP segments into frames of predictable lengths—and thus into predictable patterns of OOK signals.

Thus, a server controlling the throughput of a TCP

[8] Released by Travis Goodspeed, KK4VCZ, as `https://github.com/travisgoodspeed/goodpsk/`

[9] See `https://github.com/debanjum/Polyglots`

socket may create a signature OOK signal that, if leaked by the client, would remotely identify the client among many other, should its Ethernet cabling have any signal-leaking flaws.

## 5 Toward a Periodic Table of PHY

The previous examples, however simple, worked because of several basic injections that allow PHYs to cross-talk.

**(a)** If the original PHY uses the same class of modulation but more discrete values of frequency or phase, the lower-valued PHY may be emulatable by a higher-valued one. This works for 4FSK $\rightarrow$ 2FSK, 4PSK $\rightarrow$ 2PSK, etc.

**(b)** Generally speaking, $n$-FSK $\rightarrow$ $m$-FSK for $n > m$ is straightforward. For PSK, such injection is more complicated, due to different receiver designs.

**(c)** Most protocols allow a polyglot with OOK, due to the principles above and the simplicity of OOK. PSK $\rightarrow$ OOK is more awkward due to having to drop amplitude for phase shifts—or having to suffer the boundary effects.

**(d)** In theory, PSK $\rightarrow$ FSK might be possible, since a phase shift is the same as a very abrupt and fine frequency shift. However, this is difficult to use in practice.

Based on the above, one may imagine a "periodic table" of PHY, gathering the different traits of the modulation schemes (at least) into a single structure that aligns similar traits. Our inspiration for this is the famous Periodic Table of chemical elements. While our proposal below is far from the systemic beauty of the modern periodic table, we would like to point out that it started from Mendeleev's sketches such as Figure 7 that looked not nearly as impressive.

One could imagine a table with rows corresponding to the number of discrete values used in shift keying, and the columns corresponding to different modulation scheme families: ASK, FSK, PSK. Thus 2FSK, 4FSK, 8FSK, etc. would form a column, and so would 2PSK, 4PSK, 8PSK, etc. Connecting the cells of this table with arrows whenever the PHY of the arrow's origin can be corrupted or manipulated to appear to PHY at the arrow's head as a valid signal (whatever else that signal might be) would exhibit a regular structure. For example, emulating an $n$-PSK or $n$-FSK PHY with an $m$-PSK or an $m$-FSK one respectively for some $m > n$ would happen in the same column, with some periodicity; polyglot relations would be diagonal, etc.

Instead of the table form, however, we propose a different depiction shown in Figure 8. In it, we arrange PHY modulation schemes across the rays of three axes: amplitude $A$, frequency $\omega$, and phase $\theta$. We arrange modulations that use shifts between increasing numbers of values along the same ray.

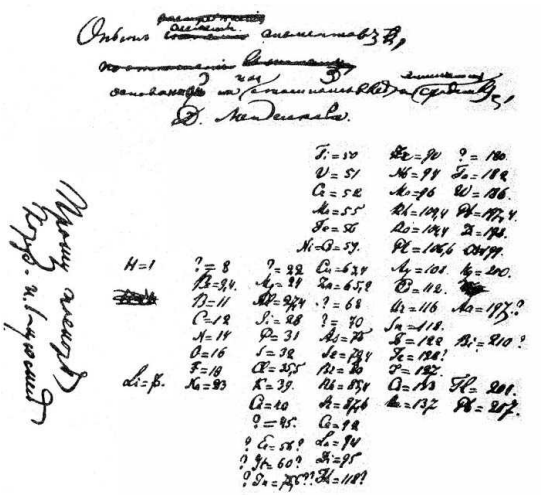

Figure 7: Dmitry Mendeleev's sketch of the periodic table, 1869

These axes are *not* orthogonal and thus not perpendicular in our depiction. For example, a rapid change in phase $\theta$ at non-zero amplitude $A$ produces frequency artifacts. In fact, modulating phase to stay in a narrow frequency band requires an amplitude envelope in $A$—although such auxiliary "modulation" of amplitude carries no information. Similar relationships between other rays exist.

The diagram also shows that modulations can be combined under certain conditions. For example, to a 2ASK scheme and a 4PSK one can relate an APSK scheme that uses 2 amplitudes and 4 phases in each, to the overall 8 points in its constellation, shown in the APSK sector of Figure 8. Such combined modulations can be thought of as arcs between the rays representing the pure amplitude, frequency, and phase modulation schemes.

These arcs fill the corresponding sectors, and there are also rays corresponding to multiples of discrete values used. Along these arcs and rays, PHYs closer to the origin can be emulated by those farther away from it.

Modulation schemes that combine discrete amplitude and frequency (AFSK), or amplitude and phase (APSK) variations to encode information populate the two sectors of the diagram. One sector remains unpopulated, since varying phase and frequency at the same time does not make for natural, stable modulations. Yet our polyglot would be in this sector. Here be dragons, and lions abound indeed![10]

We call for further exploration of this structure.

[10]The drawing of the basilisk is by Ulisse Aldrovandi, 1640.

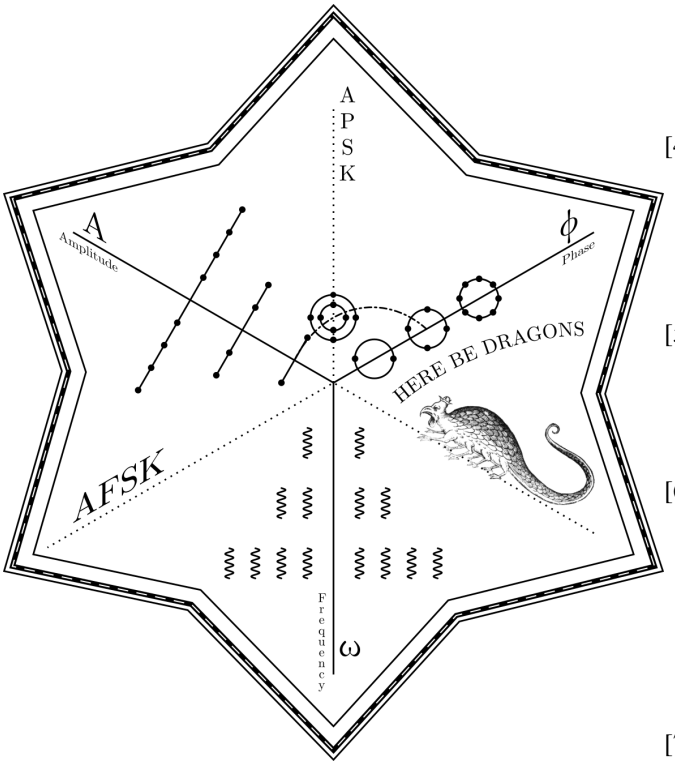


Figure 8: A PHY-riodic table of modulation schemes.

## Conclusion

No PHY is an island just because it was not specially designed to be compatible with others. Although our intuition suggests that meaningful communication between non-compatible PHYs is not possible, and that two digital radios cannot be receiving two different valid payloads at the same time from the same signal in the radio medium, simple examples show that cross-PHY communication and multi-PHY polyglots are possible, and should be looked for systematically.

## References

[1] Ange Albertini. Abusing file formats; or, Corkami, the novella. *PoC||GTFO*, 7, March 2015.

[2] Sergey Bratus, Cory Cornelius, David Kotz, and Daniel Peebles. Active behavioral fingerprinting of wireless devices. In *Proceedings of the first ACM conference on wireless network security*, WiSec '08, pages 56–61, 2008.

[3] Daniel Genkin, Lev Pachmanov, Itamar Pipman, and Eran Tromer. Stealing keys from PCs using a radio: cheap electromagnetic attacks on windowed exponentiation. In *Cryptographic Hardware and Embedded Systems–CHES 2015*, pages 207–228. Springer, 2015.

[4] Travis Goodspeed. Promiscuity is NRF24L01's duty. `http://travisgoodspeed.blogspot.com/2011/02/promiscuity-is-nrf24l01s-duty.html`, February 2011.

[5] Travis Goodspeed. Phantom boundaries and cross-layer illusions in 802.15. 4 digital radio. In *Security and Privacy Workshops (SPW), 2014 IEEE*, pages 181–184. IEEE, 2014.

[6] Travis Goodspeed, Sergey Bratus, Ricky Melgares, Rebecca Shapiro, and Ryan Speers. Packets in Packets: Orson Welles' In-Band Signaling Attacks for Modern Radios. In David Brumley and Michal Zalewski, editors, *5th USENIX Workshop on Offensive Technologies*, pages 54–61. USENIX, August 2011.

[7] Travis Goodspeed, Sergey Bratus, Ricky Melgares, Ryan Speers, and Sean W. Smith. Api-do: Tools for exploring the wireless attack surface in smart meters. In *45th Hawaii International International Conference on Systems Science (HICSS-45)*, pages 2133–2140, 2012.

[8] Ira Ray Jenkins, Rebecca Shapiro, Sergey Bratus, Ryan Speers, and Travis Goodspeed. Fingerprinting IEEE 802.15.4 devices with commodity radios. Technical report, Dartmouth Computer Science Technical Report TR2014-746, 2014.

[9] Dan Kaminsky, Len Sassaman, and Meredith Patterson. PKI Layer Cake: New Collision Attacks Against The Global X.509 CA Infrastructure. Black Hat USA, August 2009. `http://www.cosic.esat.kuleuven.be/publications/article-1432.pdf`.

[10] Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu. Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors. In *ACM SIGARCH Computer Architecture News*, volume 42, pages 361–372. IEEE Press, 2014.

[11] Markus G Kuhn and Ross J Anderson. Soft tempest: hidden data transmission using electromagnetic emanations. In *Information Hiding*, pages 124–142. Springer, 1998.

[12] Haroon Meer. The (almost) complete history of memory corruption attacks. BlackHat 2010, Aug 2010.

[13] Christina Pöpper, Nils Ole Tippenhauer, Boris Danev, and Srdjan Capkun. Investigation of Signal and Message Manipulations on the Wireless Channel. In *Proceedings of the 16th European Conference on Research in Computer Security*, ESORICS'11, pages 40–59, 2011.

[14] Mark Seaborn and Thomas Dullien. Exploiting the DRAM rowhammer bug to gain kernel privileges. BlackHat 2015, Aug 2015.

[15] László Szekeres, Mathias Payer, Tao Wei, and Dong Song. SoK: Eternal war in memory. In *2013 IEEE Symposium on Security and Privacy*, pages 48–62. IEEE, May 2013.

[16] Matthias Wilhelm, Vincent Lenders, and Jens B. Schmitt. On the Reception of Concurrent Transmissions in Wireless Sensor Networks. In *IEEE Transactions on Wireless Communications*, volume 13, pages 6756–6767. IEEE, December 2014.