Assessing the Spoofing Threat: Development of a Portable GPS Civilian Spoofer

Todd E. Humphreys, *University of Texas at Austin, Austin, TX*
Brent M. Ledvina, *Virginia Tech, Blacksburg, VA*
Mark L. Psiaki, Brady W. O'Hanlon, and Paul M. Kintner, Jr., *Cornell University, Ithaca, NY*

BIOGRAPHIES

Todd E. Humphreys is a research assistant professor in the department of Aerospace Engineering and Engineering Mechanics at the University of Texas at Austin. He will join the faculty of the University of Texas at Austin as an assistant professor in the Fall of 2009. He received a B.S. and M.S. in Electrical and Computer Engineering from Utah State University and a Ph.D. in Aerospace Engineering from Cornell University. His research interests are in estimation and filtering, GNSS technology, GNSS security, and GNSS-based study of the ionosphere and neutral atmosphere.

Brent M. Ledvina is an Assistant Professor in the Electrical and Computer Engineering Department at Virginia Tech. He received a B.S. in Electrical and Computer Engineering from the University of Wisconsin at Madison and a Ph.D. in Electrical and Computer Engineering from Cornell University. His research interests are in the areas of ionospheric physics, space weather, estimation and filtering, and GNSS technology and applications.

Mark L. Psiaki is a Professor in the Sibley School of Mechanical and Aerospace Engineering. He received a B.A. in Physics and M.A. and Ph.D. degrees in Mechanical and Aerospace Engineering from Princeton University. His research interests are in the areas of estimation and filtering, spacecraft attitude and orbit determination, and GNSS technology and applications.

Brady W. O'Hanlon received a B.S. in Electrical and Computer Engineering from Cornell University, where he continues on in the M.S./Ph.D. program. His research interests include GNSS technologies and space weather.

Paul M. Kintner, Jr. is a Professor of Electrical and Computer Engineering at Cornell University. He received a B.S in Physics from the University of Rochester and a Ph.D. in Physics from the University of Minnesota. His research interests include the electrical properties of upper atmospheres, space weather, and developing GNSS instruments for space science. He is a Fellow of the APS.

ABSTRACT

A portable civilian GPS spoofer is implemented on a digital signal processor and used to characterize spoofing effects and develop defenses against civilian spoofing. This work is intended to equip GNSS users and receiver manufacturers with authentication methods that are effective against unsophisticated spoofing attacks. The work also serves to refine the civilian spoofing threat assessment by demonstrating the challenges involved in mounting a spoofing attack.

I. INTRODUCTION

In 2001, the U.S. Department of Transportation assessed the U.S. transportation infrastructure's vulnerability to civil GPS disruption [1]. Their report, known as the Volpe report, warned that "[a]s GPS further penetrates into the civil infrastructure, it becomes a tempting target that could be exploited by individuals, groups, or countries hostile to the U.S." Among other types of interference, the report considers civil GPS spoofing, a pernicious type of intentional interference whereby a GPS receiver is fooled into tracking counterfeit GPS signals. Spoofing is more sinister than intentional jamming because the targeted receiver cannot detect a spoofing attack and so cannot warn users that its navigation solution is untrustworthy. The Volpe report noted the absence of any "off the shelf" defense against civilian spoofing and lamented that "[t]here also is no open information on ... the expected capabilities of spoofing systems made from commercial components." It recommended studies to characterize the spoofing threat: "Information on the capabilities, limitations, and operational procedures [of spoofers] would help identify vulnerable areas and detection strategies."

Seven years later, civil GPS receivers remain as vulnerable as ever to this threat. In a recent informal survey conducted by the authors, four manufacturers of high-quality GPS receivers revealed that they were aware of the spoofing vulnerability, but had not taken steps to equip their receivers with even rudimentary spoofing countermeasures. The manufacturers expressed skepticism about the seriousness of the spoofing threat and noted that countermeasures, if required, had better not be too expensive. Such attitudes invite further examination of the spoofing threat and of practical spoofing countermeasures.

Important research into spoofing countermeasures has been carried out over the last decade. The Volpe report cites an internal memorandum from the MITRE Corpora-

Preprint of the 2008 ION GNSS Conference Savanna, GA, September 16–19, 2008

tion in which the author, Edwin L. Key, appears to have examined spoofing and spoofing countermeasures in detail [2]. The memorandum recommends the following techniques for countering spoofing:

1. Amplitude discrimination
2. Time-of-arrival discrimination
3. Consistency of navigation inertial measurement unit (IMU) cross-check
4. Polarization discrimination
5. Angle-of-arrival discrimination
6. Cryptographic authentication

Techniques 1 and 2 could be implemented in software on GPS receivers, but the techniques would be effective against only the most simplistic spoofing attacks. Techniques 3-5 would be effective against some—but not all—more sophisticated attacks. In particular, angle-of-arrival discrimination, which exploits differential carrier phase measurements taken between multiple antennas, could only be spoofed by a very sophisticated coordinated spoofing attack (to be discussed in Section II). However, techniques 3-5 require additional hardware, namely, multiple antennas or a high-grade IMU, whose cost militates against their widespread adoption.

Cryptographic authentication, Keys's technique 6, has been studied in some detail in the years since the Volpe report [3–5]. GNSS researcher Logan Scott offers several levels of authentication in his 2003 ION GPS/GNSS paper [3] and urges their prompt adoption in a recent article on the subject [6]. Scott's methods are backward compatible with non-compliant GPS receivers. Spreading code authentication, which is the basis for Scott's Level 2 and 3 authentication, entails embedding messages in the GPS ranging codes and periodically authenticating these messages. Because this method effectively binds a digital signature to the ranging codes, it would render a compliant receiver practically impervious to a spoofing attack except during the short interval between reception and authentication of the embedded messages.

Unfortunately, the techniques offered by Scott all require modification of the civil GPS signal structure. For comprehensive authentication, one of the L2C, L5, or L1C signals on Block IIF and Block III GPS satellites would have to be altered to incorporate the embedded messages. Such changes appear extremely unlikely in the short term because, as one experienced observer noted, "signal definition inertia is enormous" [7]. A less effective but more practical approach over the U.S. would be to authenticate only the WAAS signal, which is managed by the U.S. Department of Transportation and the Federal Aviation Administration. Since the WAAS signal is constructed on the ground and transmitted via "bent pipe" communication spacecraft, it is more amenable to immediate modification. Even so, efforts to persuade WAAS officials to adopt spreading code authentication have so far proven fruitless (Logan Scott, private communication).

The Homeland Security Institute, a research arm of the U.S. Department of Homeland Security, has also considered the threat of civil GPS spoofing. On its website it has posted a report listing seven spoofing countermeasures [8]. (This is, incidentally, the first hit that surfaces in an internet search on "GPS spoofing" at the time of writing.) The proposed countermeasures include techniques 1, 2 and 3 from Keys's list above. Among the remaining four countermeasures are techniques that would be trivial to spoof. None of the seven techniques would adequately defend against a sophisticated spoofing attack. Nonetheless, the posting claims that its proposed techniques "should allow suspicious GPS signal activity to be detected." The authors of the present paper worry that such optimistic language in such a prominent posting will mislead many readers into believing that the spoofing threat has been adequately addressed.

The goals of the present work are to provide a refined assessment of the spoofing threat and to develop and test spoofing countermeasures that are practical and effective. The authors have concluded that to advance these goals it was necessary to go through the exercise of building a civil GPS spoofer. The process of developing a complete portable spoofer allows one to explore the range of practical spoofing techniques. By this exercise, one discovers which aspects of spoofing are hard and which are easy to implement in practice. With this information, the difficulty of mounting a spoofing attack can be more accurately assessed and receiver developers can prioritize their spoofing defenses by choosing countermeasures that are effective against easily-implementable spoofing techniques.

Software-defined GPS receivers are a natural platform for the study of civil spoofing and its effects. In a software GPS receiver, the real-time correlators, tracking loops, and navigation solver are all implemented in software on a programmable processor. The current authors have pioneered some of the efficient correlation techniques and other implementation strategies that have enabled the development of capable PC- and DSP-based software receivers [9–12]. The spoofer described in this paper is a software-defined civil GPS receiver-spoofer.

The remainder of this paper is divided into six sections. These are listed here for ease of navigation:
II: Initial Spoofing Threat Assessment
III: Receiver-Spoofer Architecture
IV: Implementation and Performance
V: Demonstration Spoofing Attack
VI: Spoofing Countermeasures Suggested by Work to Date

VII: Conclusions

II. INITIAL SPOOFING THREAT ASSESSMENT

The goal in designing secure systems is to brace a system's weakest link against foreseeable attacks [13]. One begins by identifying likely modes of attack—those that easily exploit the system's obvious vulnerabilities—and considering defenses against them. Such is the goal of this section.

Consider the spoofing threat continuum illustrated in Fig. 1. To facilitate a threat analysis, the continuum is roughly divided into simplistic, intermediate, and sophisticated spoofing attacks.

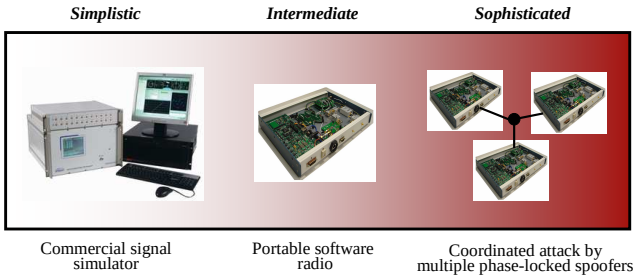

Fig. 1. The spoofing threat continuum: simplistic, intermediate, and sophisticated spoofing attacks.

A. Simplistic Attack via GPS Signal Simulator

As far as the authors are aware, all stand-alone commercial civilian GPS receivers available today are trivial to spoof. One simply attaches a power amplifier and an antenna to a GPS signal simulator and radiates the RF signal toward the target receiver. A successful attack along these lines was handily demonstrated by researchers at Argonne National Laboratories in 2002 [14] (see also the discussion at http://philosecurity.org/2008/09/07/gps-spoofing).

Despite the ease of mounting a spoofing attack with a signal simulator, there are some drawbacks. One is cost: the price of modern simulators can reach $400 k. Simulators can be rented for less than $1 k per week, which makes them accessible for short-term mischief, but long-term use remains costly. Another drawback is size. Most GPS signal simulators are heavy and cumbersome. If used in the simplest attack mode—situated close to a target receiver's antenna—a signal simulator would be challenging to plant and visually conspicuous. Of course, if the custodian of the target receiver is complicit in the spoofing attack—as is the case, for example, with the fishing vessel skipper who spoofs the onboard GPS-based monitoring unit to fish undetected in forbidden waters—the conspicuousness of the signal spoofer is irrelevant.

The menace posed by a simulator-based spoofing attack is diminished by the fact that such an attack is likely easy to detect. This is because of the difficulty of synchronizing a simulator's output with the actual GPS signals in its vicinity. An unsynchronized attack effectively acts like signal jamming, and may cause the victim receiver to lose lock and have to undergo a partial or complete re-acquisition. Such a forced re-acquisition would raise suspicion of a spoofing attack. If the unsynchronized attack somehow avoids causing loss of lock, it will nonetheless likely cause an abrupt change in the victim receiver's GPS time estimate. The victim receiver could flag jumps of more than, say, 100 ns, as evidence of possible spoofing. The spoofer can attempt to counter this defense by intentionally jamming first and then spoofing, but an extended jamming period may be required to sufficiently widen the target receiver's window of acceptance, and extended jamming is itself telltale evidence of interference.

In summary, the ease of mounting an attack via GPS signal simulator makes this attack mode relatively likely. Mercifully, detecting such an attack appears also to be easy. Of course, the mere fact that a simulator-type attack is easy to defend does not increase security. A gaping vulnerability will remain until civil GPS receivers at least are equipped with the rudimentary spoofing countermeasures required to detect a simulator-type attack.

B. Intermediate Attack via Portable Receiver-Spoofer

One of the challenges that must be overcome to carry out a successful spoofing attack is to gain accurate knowledge of the target receiver antenna's position and velocity. This knowledge is required to precisely position the counterfeit signals relative to the genuine signals at the target antenna. Without such precise positioning, a spoofing attack is easily detected.

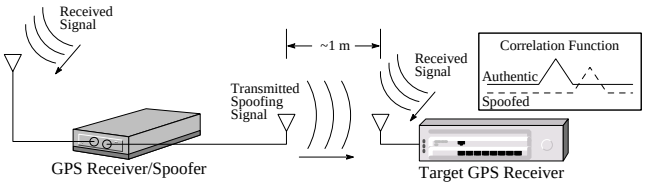

Fig. 2. Illustration of a spoofing attack via portable receiver-spoofer.

An attack via portable receiver-spoofer, portrayed in Fig. 2, overcomes this difficulty by construction. The receiver-spoofer can be made small enough to be placed inconspicuously near the target receiver's antenna. The receiver component draws in genuine GPS signals to estimate its own position, velocity, and time. Due to proximity, these apply approximately to the target antenna. Based on these estimates, the receiver-spoofer then generates counterfeit signals and generally orchestrates the spoofing attack. The portable receiver-spoofer could even be placed somewhat distant from the target receiver if the target were static

and its position relative to the receiver-spoofer had been pre-surveyed.

Each channel of the target receiver is brought under control of the receiver-spoofer as illustrated in the inset at the upper right of Fig. 2. The counterfeit correlation peak is aligned with the peak corresponding to the genuine signal. The power of the counterfeit signal is then gradually increased. Eventually, the counterfeit signal gains control of the delay-lock loop tracking points that flank the correlation peak.

As one might imagine, there are no commercially available portable receiver-spoofer devices. This of course decreases the present likelihood of the receiver-spoofer attack mode. Nonetheless, the emergence of software defined GPS receivers significantly erodes this barrier. As will be demonstrated subsequently, the hardware for a receiver-spoofer can be assembled from inexpensive off-the-shelf components. The software remains fairly sophisticated, but it would be unwise to assume it was beyond the capabilities of clever malefactors. The civil GPS signal structure is, after all, completely detailed in a publicly available interface control document [15], and entire books have been written on software-defined GPS receivers [16]. In perhaps the most worrisome scenario, anticipated in Ref. [3], the software definition of a receiver-spoofer may someday be available for download from the Internet. The expertise required to download and exercise the code would surely be within the reach of many potential malefactors.

An attack via portable receiver-spoofer could be difficult to detect. The receiver-spoofer is able to synchronize its signals to GPS time and, by virtue of its proximity to the target antenna, align the counterfeit and genuine signals. A target receiver equipped with a stable reference oscillator and a low-drift IMU (for receivers on dynamic platforms) could withstand an attack via receiver-spoofer for several hours. Eventually, however, a patient receiver-spoofer would gain undetected control by keeping its perturbations to time and position within the envelope allowed by the drift rates of the target receiver's oscillator and IMU.

The only known user-equipment-based countermeasure that would be completely effective against an attack launched from a portable receiver-spoofer with a single transmitting antenna is angle-of-arrival discrimination. With a single transmitting antenna, it would be impossible to continuously replicate the relative carrier phase between two or more antennas of an appropriately equipped target receiver.

In summary, an attack via portable receiver spoofer is not presently likely because such a device is not readily available. However, the emergence of software-defined GPS receivers increases the future likelihood of such an attack. Furthermore, this mode of attack could defeat most known user-equipment-based spoofing countermeasures.

C. Sophisticated Attack via Multiple Phase-locked Portable Receiver-Spoofers

The angle-of-arrival defense against a portable receiver-spoofer can be thwarted by a coordinated attack with as many receiver-spoofers as antennas on the target receiver. Imagine a receiver-spoofer the size of a pack of cards—small enough to mount directly atop a target antenna. The receiver-spoofer's receiving and transmitting antennas are situated respectively on the upper and lower faces of the device and are shielded to avoid self-spoofing. Now imagine several such devices sharing a common reference oscillator and communication link, with each device mounted to one of the target receiver's antennas. The angle-of-arrival defense fails under this attack scenario.

Naturally, this attack inherits all of the challenges of mounting a single receiver-spoofer attack, with the additional expense of multiple receiver-spoofers and the additional complexity that the perturbations to the incoming signals must be phase coordinated.

The only known defense against such an attack is cryptographic authentication.

In summary, an attack via multiple phase-locked portable receiver-spoofers is somewhat less likely than an attack via single portable receiver-spoofer, but may be impossible to detect with user-equipment-based spoofing defenses.

D. Target Spoofer Type

The foregoing discussion of the spoofing threat continuum suggests that a spoofing attack via GPS signal simulator poses the greatest near-term threat. However, there are known effective defenses against such an attack and these can be implemented in software on commercial GPS receivers. In contrast, an attack launched from one or more portable receiver-spoofer(s) poses the greatest long-term threat. Known user-equipment-based defenses against such attacks are few and of limited effectiveness. Accordingly, focus will be directed in this paper toward the portable receiver-spoofer attack mode. To better understand this mode, a software-defined portable receiver-spoofer has been built as a research platform.

III. RECEIVER-SPOOFER ARCHITECTURE

The software-defined receiver-spoofer that has been developed is an extension of the Cornell GRID receiver [12]. A spoofer software module and transmission hardware have been added. A top-level block diagram of the receiver-spoofer is shown in Fig. 3.

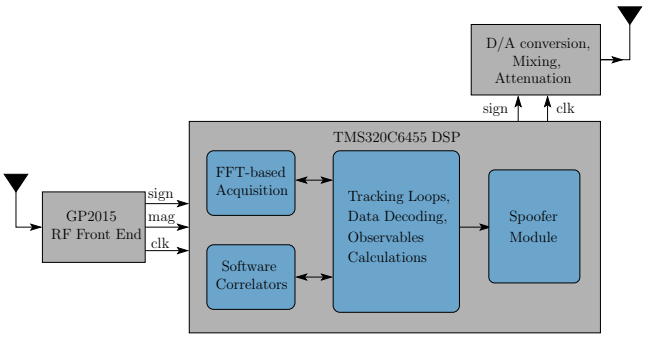

Fig. 3. Block diagram of the receiver-spoofer architecture.

A. Receiver Module

The receiver hardware of the receiver-spoofer consists of a Zarlink/Plessey GP2015 RF front end, a CPLD for signal multiplexing (not shown), and a Texas Instruments TMS320C6455 DSP. The receiver software that runs on the DSP is similar to that described in [12] except that it now includes a full navigation solution engine. The software is entirely written in natural-language C++, which facilitates code development and maintenance.

The software correlation engine, which is based on the bit-wise parallel correlation technique introduced in Refs. [9] and [11], is crucial to meeting real-time deadlines in the receiver-spoofer under the simultaneous burdens of receiver processing and spoofing. Accordingly, an overview of the bit-wise parallel technique is given here. For other details on the receiver module, the reader is directed to Ref. [12].

Figures 4, 5, and 6 are meant to facilitate explication of the bit-wise parallel correlation technique. Figure 4 depicts the standard correlation operation that occurs within any GPS receiver. The incoming signal $x(t)$ is mixed by complex multiplication with a complex local signal replica, $x_l(t)$. The product is integrated over a short interval (typically 1-20 ms) and sampled to produce the quadrature baseband components I_k and Q_k, also known as baseband accumulations.

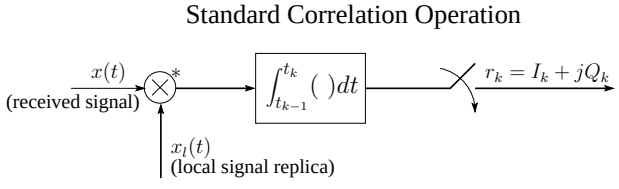

Fig. 4. Standard correlation operation. The local signal replica $x_l(t)$ is complex and $\otimes^*$ denotes complex multiplication.

Figure 5 depicts a byte-wise software implementation of the standard correlation operation. In this implementation, the individual signal samples are stored in 8-bit bytes. Because many DSPs and general-purpose CPUs are capable of performing several multiply-and-accumulate operations in parallel (e.g., 8 in high-performance TI fixed-point DSPs), the byte-wise implementation can be quite computationally efficient. However, storing the local carrier and code replica samples as bytes makes the tables in which they are packed for efficient table look-up prohibitively large for storage in on-chip (fast) memory. Furthermore, despite its computational efficiency, the byte-wise implementation is still only one-quarter to one-half as fast as the bit-wise parallel implementation when implemented on a high-performance fixed-point DSP.

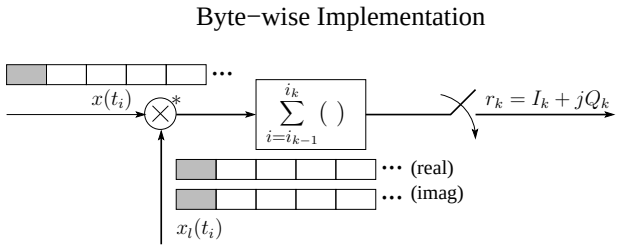

Fig. 5. Byte-wise implementation of the correlation operation. Boxes in the signal trains represent bytes, each of which stores an 8-bit signed representation of the signal x or of the complex local replica x_l. Grayed boxes represent the operands of one complex multiplication operation.

Figure 6 depicts the bit-wise parallel correlation implementation. As implemented on the receiver-spoofer, the bit-wise parallel correlation operation assumes the incoming signal and the local signal replicas are quantized to two bits—one sign and one magnitude bit. The sign and magnitude bits are packed into 32-bit words. Explicit complex multiplication is replaced by a combination of the bit-wise logical operations AND, NOR, and XOR. In effect, the bit-wise parallel method performs 32 multiply-and-accumulate operations in parallel. Importantly, storage of the local carrier replicas as bit-packed sign and magnitude words is also memory-efficient, which makes on-chip storage of the local signal replica look-up tables possible.

B. Spoofer Module

Beyond the hardware required for the GPS receiver, the receiver-spoofer requires only signal transmission hardware: a D/A converter, a frequency synthesizer and mixer for mixing to near the GPS L1 frequency, in-line attenuators, and a transmission antenna. For the present paper, no over-the-air tests were conducted (to avoid possible FCC violations); hence, the transmission hardware will not be discussed further.

The heart of the spoofer is the spoofer software module, which is shown in greater detail in Fig. 7. The main components of the spoofer module are described in the following subsections.

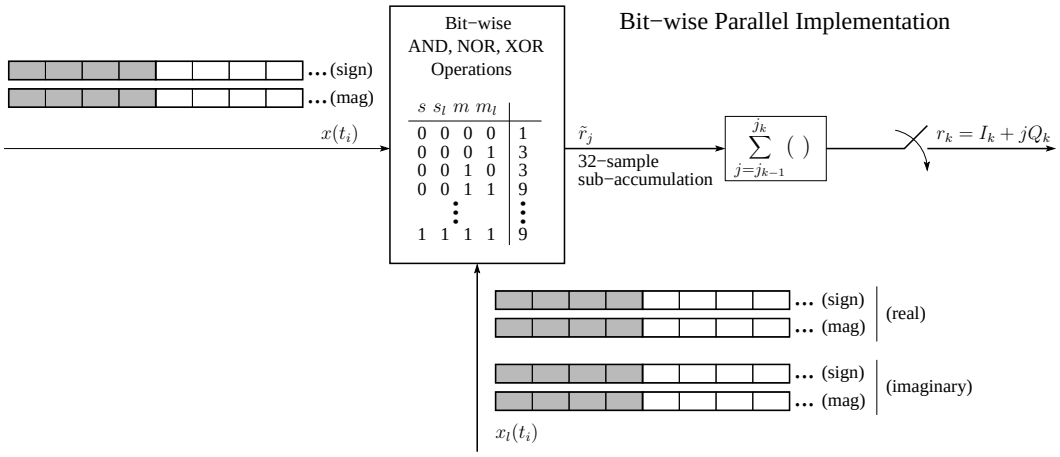

Fig. 6. Bit-wise parallel implementation of the correlation operation. Boxes in the signal trains represent 8-bit bytes. Grayed boxes represent the operands of one complex multiplication operation, which is implemented by bit-wise AND, NOR, and XOR operations.

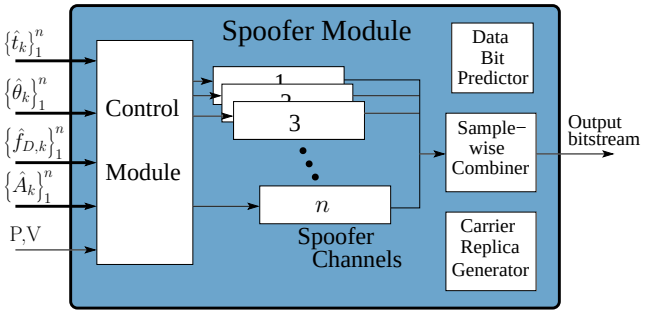

Fig. 7. Block diagram of the spoofer module.

B.1 Control Module

The spoofer's control module coordinates a spoofing attack by directing the frequency, code phase offset, and signal amplitude applied in each of n spoofing channels. Some components of the control module described below remain under development.

The control module accepts the following inputs from the receiver module: the estimates $\{\hat{t}_k\}_1^n$ of the start times of the kth C/A code period on receiver channels 1-n; the estimates $\{\hat{\theta}_k\}_1^n$ of the beat carrier phase on receiver channels 1-n at times $\{\hat{t}_k\}_1^n$; the estimates $\{\hat{f}_{D,k}\}_1^n$ of the Doppler frequency shift on receiver channels 1-n at times $\{\hat{t}_k\}_1^n$; the estimates $\{\hat{A}_k\}_1^n$ of the signal amplitudes on receiver channels 1-n at times $\{\hat{t}_k\}_1^n$; and the receiver-spoofer's current 3-dimensional position P and velocity V.

The control module orchestrates a spoofing attack in the following way. It begins by commanding n spoofer channels to generate signals with Doppler frequency offsets equal to $\{\hat{f}_{D,k}\}_1^n$ and code phases whose relative alignment is equivalent to that dictated by $\{\hat{t}_k\}_1^n$. It then applies a common-mode code phase advance to compensate for buffering delays within the receiver-spoofer. If this advance is chosen correctly, then each spoofing signal will be code-phase-aligned with its genuine-signal counterpart at the target receiver's antenna. The control module then commands an increase in the signal amplitude of one or more spoofer channels to effect lift-off of the target receiver's tracking points. This continues until all target receiver channels are presumed to be under control of the spoofer.

At this point the control module gradually leads the target receiver off its true position and time to an alternate position or time. Let $\Delta\boldsymbol{x}_D(t_k) = [\Delta v_x(t_k), \Delta v_y(t_k), \Delta v_z(t_k), \Delta\dot{b}(t_k)]^T$ be the perturbation that the control module applies to the target receiver's observed velocity and clock rate bias at receiver-spoofer time t_k. The time rate of change of the perturbation $\Delta\dot{b}(t_k)$ must be less than the expected drift rate of the target receiver's reference oscillator. Likewise, the time rate of change of the velocity perturbations $\Delta v_x(t_k), \Delta v_y(t_k)$, and $\Delta v_z(t_k)$ must be less than the accelerations that the target receiver expects, or, if the target receiver is equipped with an IMU, less than the expected uncertainty in the accelerometer bias.

To enforce $\Delta\boldsymbol{x}_D(t_k)$, the control module linearizes the standard Doppler frequency measurement model about the current receiver time, position, and velocity estimates and computes offsets to the quantities $\{\hat{f}_{D,k}\}_1^n$ that are commensurate with the perturbation $\Delta\boldsymbol{x}_D(t_k)$.

Similarly, let $\Delta\boldsymbol{x}(t_k) = [\Delta x(t_k), \Delta y(t_k), \Delta z(t_k), \Delta t(t_k)]^T$ be the perturbation that the control module applies to the target receiver's observed position and time at receiver-spoofer time t_k. $\Delta\boldsymbol{x}(t_k)$ is calculated by integrating the time history of $\Delta\boldsymbol{x}_D(t_k)$ values from some initial condition, typically $\Delta\boldsymbol{x}_D(t_k) = 0$ so that the target receiver's observed velocity and clock rate bias is initially approxi-

mately equal to its true velocity and clock rate bias. To enforce $\Delta\boldsymbol{x}(t_k)$, the control module linearizes the standard pseudorange measurement model about the current receiver time and position estimates and computes offsets to the quantities $\{\hat{t}_k\}_1^n$ that are commensurate with the perturbation $\Delta\boldsymbol{x}(t_k)$.

Following the above strategy, the control module can, as gradually as necessary, misdirect the target receiver's observed position and time.

The spoofer control module currently makes no attempt to align the beat carrier phases of its output signals with those of the received GPS signals, and so the phase values $\{\hat{\theta}_k\}_1^n$ are currently discarded. More sophisticated future versions of the receiver-spoofer will likely make use of these phase values.

B.2 Spoofer Channels

Each of the n spoofer channels is configured to correspond to one of the n authentic GPS signals that the receiver module tracks. The signal generated by the nth spoofer channel can be modeled as

$$x_n(\tau_i) = A_n(\tau_i)d_n(\tau_i)C_n(\tau_i - t_{n,k}) \times Q\left\{\sin\left[2\pi f_{IF}\tau_i + \theta_n(\tau_i)\right]\right\} \quad (1)$$

$$\dot{\theta}_n(\tau_i = t_{n,k}) = f_{D,n,k} \quad (2)$$

where $x_n(\tau_i)$ is the ith sample of the signal, τ_i is the time of the ith sample, $A_n(\tau_i)$ is the control-module-commanded amplitude at τ_i, $d_n(\tau_i)$ is the data bit value that applies at τ_i, $C_n(\tau_i - t_{n,k})$ is the C/A code chip value that applies at τ_i, $t_{n,k}$ is the control-module-commanded start time of the kth C/A code period, $Q\{\cdot\}$ is a 2-bit quantization function, f_{IF} is the intermediate frequency, $\theta_n(\tau_i)$ is the beat carrier phase at τ_i, and $f_{D,n,k}$ is the control-module-commanded Doppler frequency shift at time $t_{n,k}$. The C/A code function $C_n(\tau)$ can be further represented as

$$C_n(\tau) = \sum_{j=1}^{1023} c_{n,j}\Pi_{T_c}(\tau - jT_c) \quad (3)$$

and the data bit function $d_n(\tau)$ as

$$d_n(\tau) = \sum_{j=-\infty}^{\infty} d_{n,j}\Pi_{T_d}(\tau - jT_d) \quad (4)$$

where $\{c_{n,1}, c_{n,2}, ..., c_{n,1023}\}$ and $\{d_{n,j}, d_{n,j+1}, ...\}$ are the unique C/A code chip sequence and navigation data bit sequence corresponding to the GPS satellite whose signal is being emulated on the nth spoofer channel, T_c and T_d are the duration of one C/A code chip and one navigation data bit, and $\Pi_T(\tau)$ is the usual rectangular support function equal to unity over $0 \le \tau < T$ and zero otherwise.

To generate the C/A code samples $\{C_n(\tau_i)\}, i = 1, 2, ...,$ the spoofer channels make use of the same bit-packed C/A code replicas that are employed for signal correlation in the receiver module, which are stored in large look-up tables. However, to generate the samples of the quantized carrier replica

$$Q\left\{\sin\left[2\pi f_{IF}\tau_i + \theta_n(\tau_i)\right]\right\}, \quad i = 1, 2, 3, ... \quad (5)$$

the spoofer channels cannot exploit the same bit-packed carrier replicas that are used for signal correlation in the receiver. This is because, to minimize on-chip memory requirements, the receiver's carrier replicas all begin at the same phase value and have only a coarse 175-Hz frequency resolution. The receiver compensates for these factors by performing a rotational "fix-up" on the in-phase and quadrature accumulation values. Unfortunately, such a scheme is unworkable for generating the sampled carrier replicas in the spoofer channels because anything less than precise phase and frequency control over the carrier replicas would potentially alert a target receiver to a spoofing attack. Consequently, it was necessary to develop a carrier replica generator more capable than that used in the receiver module.

B.3 Carrier Replica Generator

Two requirements drove the carrier replica generator design: precision and efficiency. Regarding precision, to evade detection the generator must be able to set the initial phase of a carrier replica segment to within approximately one degree and the Doppler frequency offset over the segment to within approximately 1 Hz. Regarding efficiency, to meet real-time deadlines the generator would have to be capable of generating a replica segment T_s seconds long in less than $T_s/30$ seconds.

A generator was developed that met these requirements. An overview of the generator is given here. Details are provided in Ref. [17].

A quantized sampled carrier replica can be represented in bit-wise parallel format as a block of 32-bit words. In the simplest case, the carrier replicas are one-bit quantized with 0 and 1 respectively representing the values -1 and 1. The carrier replica generator can be configured to generate 1- to 4-bit-quantized samples. Two-bit quantization was chosen for implementation within the spoofer, with one bit representing the sign and the other representing the magnitude of the signal. The choice of 2-bit quantization balanced a tradeoff between efficiency and the amount of quantization noise introduced into the final linear combination of the spoofer channel outputs.

The carrier replicas are sampled at a rate $f_s > 2f_{IF}$ Hz as shown for the minimum and maximum Doppler frequency shifts in Fig. 8. The key observation that makes real-time generation of the carrier replicas possible is the following: There is little diversity in the 32-bit words that result

from packing 32 samples of quantized carrier replicas over a ±10-kHz range of Doppler frequency offsets and 2π radians of carrier phase. This is another way of saying that the information content of the quantized sampled carrier replicas is low, which is to be expected.

Figure 8 illustrates this concept by showing a case with a sampling frequency $f_s = 5.714$ MHz, an intermediate frequency $f_{IF} = 1.405$ MHz, and a Doppler frequency range of ±10kHz. This Doppler frequency range covers the expected range of Doppler shifts seen by a terrestrial GPS receiver, with ∼5 kHz of margin for receiver clock rate error. The sampling and intermediate frequencies are typical for civil GPS applications. Over the interval shown in Fig. 8, the total number of cycles for the two signals, whose initial phases are aligned, differs by less than 1/8 of a cycle. When sampled and 2-bit quantized into the sign (s) and magnitude (m) bits that run along the bottom of each frame, the resultant carrier replicas have the same sign-bit history and only 10 different magnitude bits. This indicates that the sampled carrier replicas covering a reasonable Doppler shift frequency range are primarily a function of the initial phase offset for each 32-bit word. This observation remains true whenever $f_{IF} < f_s$ and $f_{D,mabs} << f_{IF}$, where $f_{D,mabs}$ is the maximum absolute value of the Doppler frequency shift.

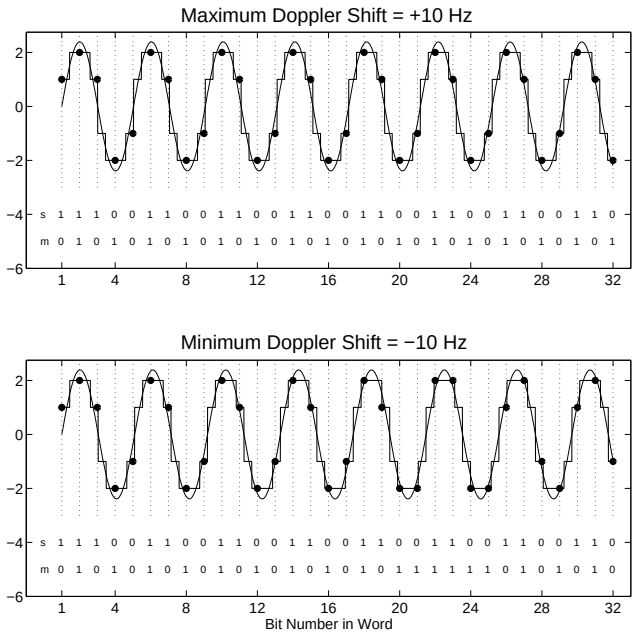

Fig. 8. Two-bit quantization of the local carrier replica at the maximum and minimum Doppler frequency shifts.

The low information content of the sampled carrier replicas makes them amenable to tabular storage and efficient retrieval. Two tables are required, one each for the sign and magnitude bits. Let $i_f \in \{0, 1, ..., N_f - 1\}$ and $i_\theta \in \{0, 1, ..., N_\theta - 1\}$ represent the respective indices into the frequency and phase dimensions of the tables. For each carrier replica segment (typically 1-ms long), a single frequency index is calculated as

$$i_f = \text{round}\left(\frac{f_D - f_{D,min}}{f_{D,max} - f_{D,min}} N_f\right) \tag{6}$$

where f_D is the exact desired frequency and $f_{D,min}$ and $f_{D,max}$ are the minimum and maximum Doppler frequency shifts. The phase index i_θ is different for each of the 32-bit words that are strung together to compose the carrier replica segment. Let τ_k be the time offset of the midpoint of the kth word in the segment relative to the time of the first sample in the segment. The phase at the midpoint of the kth word is calculated as

$$\theta_{mid,k} = \text{mod}\left[2\pi(f_{IF} + f_D)\tau_k + \theta_0\right] \tag{7}$$

where θ_0 is the phase of the first sample in the segment, and the modulo operation is modulo 2π. Finally, the phase index of the kth word is calculated as

$$i_{\theta_k} = \text{round}\left(\frac{\theta_{mid,k}}{2\pi} N_\theta\right) \tag{8}$$

To meet precision requirements, the number of indices into the frequency and phase dimensions of the tables were set respectively to $N_f = 32$ and $N_\theta = 256$. With this table size, the table-generated carrier replicas are not significantly different from carrier replicas generated by applying the exact phase and frequency values using double-precision computations. The sign and magnitude tables occupy a total of 64 kB in on-chip memory.

B.4 Data Bit Predictor

The GPS L_1 navigation data bit sequence $\{d_{n,j}, d_{n,j+1}, ...\}$ required by the nth spoofer channel is most easily generated in one of two ways. The simplest approach is to pass data bits to the spoofer channels as soon as they can be reliably read off the incoming GPS signals. Naturally, this approach results in a delay in the arrival time of the spoofing data bit as compared to that of the true data bit at the target receiver's antenna. The delay is most conveniently made an integer number of 1-ms C/A code intervals. Clearly, such a delay is undesirable in a spoofer because a target receiver could be designed to watch for such a delay and thereby detect a spoofing attack.

The second approach is to predict the data bits based on knowledge of the bit structure and a recent bit observation interval. This is the function of the receiver-spoofer's data bit predictor. This method relies on the fact that the GPS navigation message has a 12.5-minute period and remains nearly perfectly predictable for a period of two hours. In fact, the almanac component of the 12.5-minute data block is refreshed by the GPS Control Segment only once per day, and the remaining data—the individual satellite

ephemeris data—can be observed in less than one minute. There are data bit segments within the TLM word of the navigation message that are unpredictable on a regular basis. However, these segments are also unpredictable for the target receiver (in the absence of external data bit aiding). Therefore, the spoofer can simply fill the unpredictable data bit segments with arbitrary data bits and adapt the parity bits and HOW word polarity accordingly.

Discrepancies have been observed between the almanac data of Block IIA and later satellites. For example, the least significant bits of particular ephemeris parameters can differ. This is believed to be a rounding error in early satellites. These discrepancies cause problems with data bit prediction for Block IIA satellites. The GPS control segment has been alerted to this and is taking corrective measures. Meanwhile, the spoofer module's data bit predictor keeps two copies of almanac data: one for Block IIA and one for later satellites.

During a spoofing attack, rising GPS satellites pose a challenge for the data bit predictor; indeed, for the entire receiver-spoofer. The receiver-spoofer must prevent the target receiver from acquiring bit lock on the new signal until the data bit predictor has a chance to observe the new satellite's ephemeris data. This could be done by transmitting a spoofing signal with arbitrary data bits whose boundaries change sporadically by an integer number of C/A code periods.

B.5 Sample-wise Combination of Spoofer Channel Output Signals

Combination of the bit-packed signals generated in each of the spoofer channels is performed sample-by-sample. The ith sample from the nth spoofer channel is weighted by $A_n(\tau_i)$ and summed with the corresponding samples from the other spoofer channels, each weighted appropriately. While computationally expensive, sample-wise operations are necessary to generate a combined signal that represents a quantized superposition of the individual spoofing signals with correct relative amplitudes. The composite signal is then re-quantized to 1 or 2 bits before being loaded into the output circular buffer. Re-quantization of the composite signal introduces additional signal distortion, which decreases the carrier-to-noise ratio of each component signal. For 1-bit re-quantization, which is the current configuration, the signal distortion is tolerable until more than 8 spoofing signals are combined. More precisely, 1-bit re-quantization can sustain no more than 8 equal-amplitude component signals at a carrier-to-noise ratio of $C/N_0 = 48$ or higher.

IV. IMPLEMENTATION AND PERFORMANCE

The software-defined receiver-spoofer has been implemented on the Cornell GRID receiver platform shown in Fig. 9. The core processor on the latest version of the GRID receiver is a 1.2 GHz Texas Instruments TMS320C6455 DSP. Both the receiver and spoofer software modules run on the same processor.

Fig. 9. The Cornell GRID receiver – the hardware platform for the receiver-spoofer.

When tuned for efficiency, the receiver-spoofer meets real-time deadlines with computational resources to spare. At full capability, the receiver-spoofer tracks 12 GPS L_1 C/A signals and simultaneously generates 12 spoofing signals, in addition to performing a 1-Hz navigation solution and continuous background acquisition. As mentioned in Section III-B.5, the 1-bit re-quantization of the composite spoofing signal limits the spoofer module practically to 8 component signals. Future versions of the receiver-spoofer may trade computational resources for 2-bit re-quantization, permitting more than 8 component spoofing signals.

The marginal computational demands of each tracking and spoofing channel are respectively 1.2% and 4% of the DSP, the latter value reflecting the high computational cost of carrier replica generation and sample-wise signal combination within the spoofer module.

The core Cornell GRID receiver software is the product of hundreds of developer-hours of work. Developing the spoofer module and extending the core GRID receiver software to include it required a team of three experienced developers working approximately 40 hours apiece, or approximately three developer-weeks. The hardware components of the receiver-spoofer platform shown in Fig. 9 are all off-the-shelf components whose total cost is approximately $1500.

V. DEMONSTRATION SPOOFING ATTACK

The following method was devised for demonstrating a spoofing attack without actually transmitting RF signals at the GPS L_1 frequency over the air, which would have violated FCC restrictions on transmitting in a protected band. An interval of digitized authentic GPS L_1 C/A

code data sampled at 5.7 MHz was stored to disk. The data were input to the receiver-spoofer, which tracked the 6 GPS signals present, generated corresponding spoofing signals, and combined these into a 1-bit quantized output bitstream. The output bitstream was then combined with the original data by interleaving, and the resulting bitstream was input to a Cornell GRID receiver acting as target receiver. A schematic of the demonstration setup is given in Fig. 10.

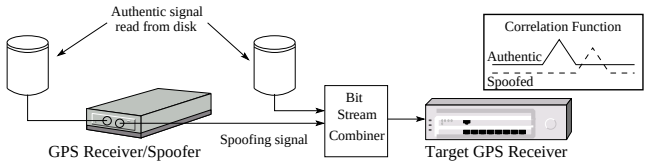

Fig. 10. The "bit combination" framework for demonstrating a spoofing attack.

The receiver-spoofer accurately reproduced the code phase, frequency, data bit values, and relative amplitude of all 6 GPS L_1 signals present. The spoofing signals' carrier phases, while not designed to match those of the genuine signals, were continuous across accumulation intervals as intended.

To enable observation of the spoofing attack, the target GRID receiver was augmented with correlator taps at 81 different 0.2-chip offsets about the prompt tap, which is nominally aligned with the incoming signal. The amplitude time history from each correlator tap can be combined to produce "footage" of the spoofing attack from the perspective of the individual channels.

Figure 11 shows a sequence of frames depicting the attack on one of the channels. The attack lasts approximately 30 seconds. Each successive panel represents a snapshot of the 81 taps' amplitudes at roughly 6-second intervals. The three red dots represent the delay-lock loop's tracking points, which continuously attempt to align themselves so that the center point is maximized and the flanking points are equalized. The first (top) frame shows the tracking points nicely aligned on the genuine signal's correlation peak, while the counterfeit signal's peak approaches furtively from the right. Of course, in a typical spoofing attack, the counterfeit peak would simply be initially aligned with the genuine peak and initially smaller than the counterfeit peak in the top panel; its approach from the right and large size in the present case is merely for clarity of presentation.

After the spoofed peak aligns with the genuine one, its signal power is gradually increased until it begins to control the tracking points. Eventually, the counterfeit peak drags the tracking points off to the left of the true peak. In the lower two panels of Fig. 11, the true peak appears to drift

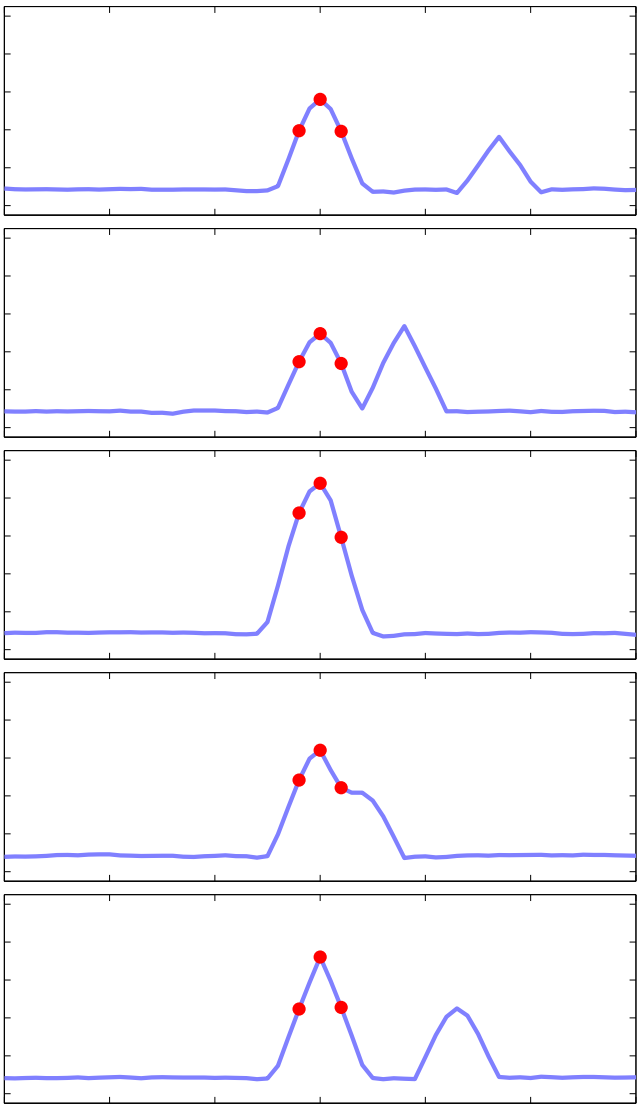

Fig. 11. A sequence of frames (from top to bottom) showing a successful single-channel spoofing attack.

off towards the right because the counterfeit peak has hijacked the 81 taps of the figure's image zone, which are tied to the victim receiver's tracking points, and it drags them all leftward relative to the true peak. A sophisticated spoofing attack will attempt right-to-left, or late-to-early, tracking lift-off wherever possible so as to disguise the attack as multipath.

Figure 12 illustrates the attack from the perspective of the baseband phasors in the complex plane. In the present version of the receiver-spoofer, no attempt is made to phase-align the authentic and spoofing signals. Consequently, a sign change in the data bit stream is possible as the spoofing phasor's amplitude gradually increases and the target receiver's phase-lock loop eventually transitions from tracking the authentic phasor to tracking the spoofing phasor. However, the rotational rates of the two phasors, ω_a

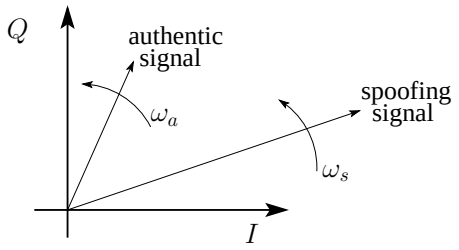

Fig. 12. The authentic and spoofing baseband phasors with respective rotational rates of ω_a and ω_s on the complex I-Q plane.

and ω_s in Fig. 12, should be nearly equivalent. From Fig. 12 it should be apparent that if a receiver-spoofer were capable of phase-aligning with a genuine signal, it could, by transmitting the exact difference between a desired spoofing signal and the true signal at the target antenna, simultaneously produce a spoofing phasor and suppress the authentic phasor. When combined with data bit prediction, such an attack could be impossible to detect relying solely on user-equipment-based defenses.

VI. SPOOFING COUNTERMEASURES SUGGESTED BY WORK TO DATE

Three spoofing countermeasures have been suggested by work to date. Two of these, both software-defined user-equipment-based defenses, are presented here. These can be thought of as additions to Keys's five user-equipment-based countermeasures presented in Section I. The third method, a promising low-impact cryptographic technique, will be disclosed in a separate publication. Neither of the user-equipment-based defenses discussed below is spoof-proof; however, each is straightforward to implement and increases the difficulty of mounting a successful spoofing attack.

A. Data Bit Latency Defense

The data bit latency defense is premised on the difficulty, discussed in Section III-B.4, of re-transmitting the GPS data bits in real time. The alternative, data-bit prediction, is itself somewhat challenging and is vulnerable to detection at the 2-hour ephemeris update boundaries and when a GPS satellite rises above the horizon.

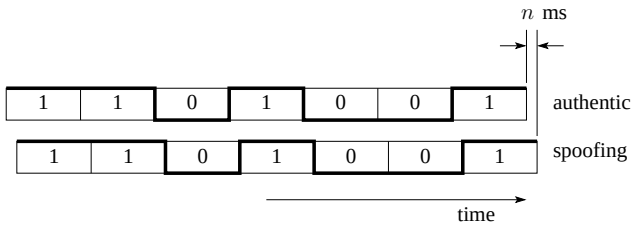

Fig. 13. Illustration of the likely latency of the spoofing data bit stream compared to the authentic data bit stream.

Figure 13 illustrates the latency between the spoofing and authentic data bit streams that would arise in the absence of data bit prediction. To detect this condition, the target receiver has only to continuously monitor bit lock. In other words, the receiver looks for a data bit sign change between consecutive accumulations at the C/A code-length interval. If a sign change is detected at other than an expected data bit boundary, then the target receiver raises a flag. Except in unusual circumstances, such as low signal power or ionospheric scintillation, a raised flag betrays a spoofing attack. The data bit latency defense has been implemented and validated on a modified Cornell GRID receiver.

Besides by data bit prediction, a spoofer can attempt to counter the data bit latency defense by jamming until the target receiver loses bit lock and then spoofing during re-acquisition. However, as with the time discrepancy defense mentioned in Section II-A, an extended jamming period may be required to sufficiently widen the target receiver's window of acceptance, and extended jamming is itself telltale evidence of interference.

B. Vestigial Signal Defense

The vestigial signal defense is premised on the difficulty of suppressing the authentic signal after successful lift-off of the delay-lock loop tracking points. To suppress the authentic signal, a spoofer must transmit the difference between a desired spoofing signal and the true signal at the target antenna, as discussed in Section V. Construction of an effective suppressor signal requires knowledge to within roughly 1/8 of a cycle of each authentic signal's carrier phase at the phase center of the target antenna. Such precise knowledge of carrier phase implies cm-level knowledge of the 3-dimensional vector between the target antenna and the transmitter phase centers. This would be challenging except in circumstances where the receiver-spoofer could be placed in the immediate proximity of the target antenna phase center.

Absent an effective suppressor signal, a vestige of the authentic GPS signal will remain in the input to the target receiver. Soon after lift-off of the delay-lock loop tracking points, the vestige may be well disguised as multipath, but its persistence and distance from the spoofed correlator peak will eventually distinguish the two effects.

To detect the vestigial authentic signal, the target receiver employs the following software-defined technique. First, the receiver copies the incoming digitized front-end data into a buffer used only for vestigial detection. Next, the receiver selects one of the GPS signals being tracked and removes this signal from the data in the buffer. This is the same technique used to remove strong signals in combating the near/far problem in spread spectrum multiple access systems, including GPS [18]. Once the tracked signal has been removed, the receiver performs acquisition for the same signal (same PRN identifier) on the buffered data.

These steps are repeated for the same GPS signal and the results are summed non-coherently until a probability of detection threshold is met for some assumed C/N_0 value and some desired probability of false alarm. If a significant vestigial signal is present in the data, this technique will reveal it.

VII. CONCLUSIONS

The deepening dependence of the civil infrastructure on GPS and the potential for financial gain or high-profile mischief makes civil GPS spoofing a gathering threat. The software-defined receiver-spoofer that has been developed and is described in this paper demonstrates that it is straightforward to mount a spoofing attack that would defeat most known user-equipment-based spoofing countermeasures. Moreover, it appears that nothing short of cryptographic authentication can guard against a sophisticated spoofing attack.

With the addition of each modernized GNSS signal (e.g., GPS L2C, L5, Galileo, and Glonass), the cost of mounting a spoofing attack rises markedly, and would quickly exceed the capabilities of the GPS L_1 civil spoofer demonstrated here. Nonetheless, faster DSPs or FPGAs would make multi-signal attacks possible. What is more, there will remain many single-frequency L_1 C/A code receivers in critical applications for years to come.

It is imperative that more research and funds be devoted to developing and testing practical and effective user-equipment-based civil GPS spoofing countermeasures such as the data bit latency defense and the vestigial signal defense introduced in this paper. Further research into cryptographic authentication methods should also be pursued. Officials in the U.S. Department of Transportation, the Federal Aviation Administration, and the Department of Homeland Security should be persuaded to consider the perils of civil GPS spoofing and to oversee development and adoption of effective countermeasures. Commercial manufacturers of GPS user equipment should be persuaded to adopt at least the rudimentary spoofing countermeasures mentioned in this paper and in the references.

In conclusion, consider the following two "security maxims" advanced by the Vulnerability Assessment Team at Argonne National Laboratory [19]. The first maxim certainly applies to civil GPS spoofing. One can only hope that the second maxim does not.

Yippee Maxim: There are effective, simple, & low-cost countermeasures (at least partial countermeasures) to many vulnerabilities.

Show Me Maxim: No serious security vulnerability, including blatantly obvious ones, will be dealt with until there is overwhelming evidence and widespread recognition that adversaries have already catastrophically exploited it. In other words, "significant psychological (or literal) damage is required before any significant security changes will be made."

ACKNOWLEDGMENTS

The Cornell GRID receiver development has been funded under ONR grant N00014-04-1-0105.

References

[1] "Vulnerability assessment of the transportation infrastructure relying on the Global Positioning System," Tech. rep., John A. Volpe National Transportation Systems Center, 2001.

[2] Key, E. L., "Techniques to Counter GPS Spoofing," Internal memorandum, MITRE Corporation, Feb. 1995.

[3] Scott, L., "Anti-spoofing and authenticated signal architectures for civil navigation systems," *Proc. ION GPS/GNSS 2003*, Institute of Navigation, Portland, Oregon, 2003, pp. 1542–1552.

[4] Hein, G., Kneissi, F., Avila-Rodriguez, J.-A., and Wallner, S., "Authenticating GNSS: Proofs against spoofs, Part 1," *Inside GNSS*, July/August 2007, pp. 58–63.

[5] Hein, G., Kneissi, F., Avila-Rodriguez, J.-A., and Wallner, S., "Authenticating GNSS: Proofs against spoofs, Part 2," *Inside GNSS*, September/October 2007, pp. 71–78.

[6] Scott, L., "Location Assurance," *GPS World*, Vol. 18, No. 7, 2007, pp. 14–18.

[7] Stansell, T., "Location Assurance Commentary," *GPS World*, Vol. 18, No. 7, 2007, pp. 19.

[8] Warner, J. S. and Johnston, R. G., "GPS spoofing countermeasures," Dec. 2003, http://www.homelandsecurity.org/bulletin /DualBenefit/warner_gps_spoofing.html.

[9] Ledvina, B. M., Cerruti, A. P., Psiaki, M. L., Powell, S. P., and Kintner, Jr., P. M., "Performance Tests of a 12-Channel Real-Time GPS L1 Software Receiver," *Proceedings of ION GPS 2003*, Institute of Navigation, Portland, OR, 2003.

[10] Ledvina, B. M., Psiaki, M. L., Powell, S. P., and Kintner, Jr., P. M., "Real-Time Software Receiver Tracking of GPS L2 Civilian Signals using a Hardware Simulator," *Proceedings of ION GNSS 2005*, Institute of Navigation, Long Beach, CA, Sept. 2005.

[11] Ledvina, B. M., Psiaki, M. L., Powell, S. P., and Kintner, Jr., P. M., "Bit-Wise Parallel Algorithms for Efficient Software Correlation Applied to a GPS Software Receiver," *IEEE Transactions on Wireless Communications*, Vol. 3, No. 5, Sept. 2004.

[12] Humphreys, T. E., Ledvina, B. M., Psiaki, M. L., and Kintner, Jr., P. M., "GNSS Receiver Implementation on a DSP: Status, Challenges, and Prospects," *Proceedings of ION GNSS 2006*, Institute of Navigation, Fort Worth, TX, 2006.

[13] Ferguson, N. and Schneier, B., *Practical Cryptography*, Wiley, 2003.

[14] Warner, J. S. and Johnston, R. G., "A simple demonstration that the Global Positioning System (GPS) Is Vulnerable to Spoofing," *Journal of Security Administration*, 2003.

[15] Anon., "ICD-GPS-200C: Navstar GPS Space Segment/Navigation User Interfaces," Tech. rep., ARINC Research Corporation, 2003, http://www.losangeles.af.mil/library /factsheets/factsheet.asp?id=9364.

[16] Borre, K., Akos, D., Bertelsen, N., Rinder, P., and Jensen, S., *A Software-defined GPS and Galileo Receiver: A Single-frequency Approach*, Birkhäuser, 2007.

[17] Ledvina, B. M., "Real-Time Generation of Bit-Packed Over-Sampled Carrier Replicas," 2008, in preparation.

[18] Johannesson, R. J., *Cross-correlation mitigation in GPS signal acquisition for a real-time software receiver*, Master's thesis, Cornell University, 2007.

[19] Johnston, R. G., "Physical security maxims," http://www.schneier.com/blog/archives/2008/09 /security_maxims.html.