A Proposed Navigation Message Authentication Implementation for Civil GPS Anti-Spoofing

Kyle D. Wesson, Mark P. Rothlisberger, and Todd E. Humphreys
The University of Texas at Austin

BIOGRAPHIES

Kyle D. Wesson is pursuing a Ph.D. in the Department of Electrical and Computer Engineering at The University of Texas at Austin. He received his B.S. in Electrical and Computer Engineering from Cornell University. He is a member of the UT Radionavigation Laboratory and the Wireless Networking and Communications Group. His research interests include GNSS security and interference mitigation.

Mark P. Rothlisberger received a Ph.D. in Mathematics from the University of Texas at Austin. He received a B.A. in Mathematics and Russian Literature from Williams College. His primary research area is Analytic Number Theory, but is also interested in applied mathematics.

Todd E. Humphreys is an assistant professor in the department of Aerospace Engineering and Engineering Mechanics at the University of Texas at Austin and Director of the UT Radionavigation Laboratory. 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-based study of the ionosphere and neutral atmosphere, and GNSS security and integrity.

ABSTRACT

A specific implementation of navigation message authentication (NMA) for civil GPS anti-spoofing is proposed. The notion of GNSS signal authentication is defined in probabilistic terms. This work proposes a practical, backward-compatible NMA strategy for the civil L2/L5 civil navigation (CNAV) message. The proposal is sufficiently detailed to facilitate near-term implementation of security-hardened civil GPS.

I. INTRODUCTION

In the decade since Selective Availability was discontinued in 2000, civil technologies based on the Global Positioning System (GPS) have become ubiquitous and the GPS service has easily achieved the stated goal of the new policy regime, which is to “encourage acceptance and integration of GPS into peaceful civil, commercial, and scientific applications worldwide; and to encourage private sector investment in and use of U.S. GPS technologies and services” [1]. Also over the past decade, the concept of national security has evolved from a focus on protecting military and critical government resources to a broader ambit that includes the protection of vital elements of civilian and commercial infrastructure. Civil GPS is a critical component of the national infrastructure; hence, GPS security is a matter of national security.

In 2001, the U.S. Department of Transportation published a report assessing the vulnerability of the U.S. transportation infrastructure to disruption of civil GPS [2]. Known as the Volpe report, it highlighted the threats posed by spoofing and meaconing attacks—methods by which a victim GPS receiver is deceived into tracking counterfeit GPS signals. At the time, the open literature contained little research on such attacks and possible countermeasures. Accordingly, the report recommended further study of GPS spoofing and development of civil GPS anti-spoofing techniques. Global Navigation Satellite System (GNSS) security research over the past decade has made much progress toward this goals [3–13].

It is convenient to distinguish cryptographic spoofing defenses, which rely on secret keys that encrypt or digitally sign components of the broadcast signals, from non-cryptographic defenses, which do not depend on encryption or digital signatures. Among non-cryptographic defenses, the multi-antenna defense [10] appears to be one of the strongest, although it remains vulnerable to the coordinated spoofing attack explored in [9]. This defense requires two or more antennas spaced by an appreciable fraction of the approximately 20-cm GPS signal wavelength, which would tend to increase receiver cost, weight, and size. As a result, the multi-antenna defense is unlikely to be widely adopted by commercial GPS manufacturers. This is also true of other non-cryptographic defenses involving inertial measurement units or other hardware, which would exceed the cost, mass, or size constraints of a broad range of applications.

Cryptographic spoofing defenses are attractive because they offer significant protection against spoofing relative to the additional cost and bulk required for implementation. While it must be conceded that no anti-spoofing technique is impervious to the most sophisticated attacks, a cryptographic defense significantly raises the bar for a successful attack and can be combined with non-cryptographic spoofing defenses for better security than either category could offer separately.

Several civil GPS cryptographic spoofing defenses have been proposed whose implementation would require fundamental changes to the legacy GPS signal structure (e.g., [3, 4, 7]). These defenses are unlikely to be implemented over the next decade given the static nature of GPS signal definitions [14].

A growing literature suggests navigation message authentication (NMA) is a practical basis for civil GPS signal authentication [3, 6, 7, 12, 15]. In NMA, the low-rate navigation message is encrypted or digitally signed, allowing a receiver to verify that the GPS Control Segment generated the data. NMA could be implemented without fundamental changes to the GPS Interface Specification by exploiting the extensibility of the modern GPS civil navigation (CNAV) messaging format. Moreover, NMA has been proposed for implementation in the European Galileo GNSS [5, 16].

Strictly speaking, NMA only authenticates the navigation message. Reference [17], which considers NMA for civil GPS anti-spoofing, recognizes this fact and further concludes that NMA is not useful for authenticating the underlying civil GPS signal. Contrary to Ref. [17], the combination of this paper and the statistical test recently developed in Ref. [13] demonstrates that NMA can in fact offer comprehensive civil GPS signal authentication if it is paired with timing authentication based on statistical hypothesis tests.

The present work offers four main contributions beyond those given in [3, 5–7, 15, 16]. First, it develops a general probabilistic interpretation of GNSS signal authentication that combines cryptographic code origin authentication with code timing authentication based on statistical hypothesis tests. Second, it identifies sensible design criteria for civil GPS signal authentication and, third, applies this framework to evaluate several proposed candidate authentication strategies. Finally, it proposes a specific cryptographic signal authentication implementation for civil GPS that meets the design criteria and is packaged for immediate adoption. The following sections are organized around these contributions, followed by conclusions.

II. SECURITY-ENHANCED GNSS SIGNAL AUTHENTICATION

Signal authentication, the topic of this paper, and message authentication, such as is used to sign data transmitted across the Internet, can be distinguished from one another by the models employed to describe their security. Message authentication security is predicated on the computational infeasibility of performing a brute-force search for the secret key used to sign the original message, or of reversing a so-called one-way function to discover the key [18]. While it is true that this assumed computational infeasibility can be couched in probabilistic terms (e.g., the probability that over the next 30 years a weakness will be found in a certain one-way hash function), such language is seldom used, either because the probabilities involved are too subjective or too small to be meaningful.

In contrast to message authentication, the security of signal authentication is much weaker and demands a probabilistic model, as described in this section.

A. Generalized Model for Security-Enhanced GNSS Signals

Current and proposed security-enhanced GNSS signals can be represented by a simple model from the perspective of a GNSS receiver. Let the signal exiting the radio frequency (RF) front-end of a GNSS receiver after having been downmixed and sampled be modeled as:

$$Y_k = w_k c_k \cos(2\pi f_{IF} t_k + \theta_k) + N_k \tag{1a}$$

$$= w_k s_k + N_k \tag{1b}$$

Here, at sample index k, w_k is a ±1-valued security code with chip length T_w, c_k is a known ±1-valued spreading (ranging) code with chip length T_c, f_{IF} is the intermediate value of the downmixed carrier frequency, θ_k is the beat carrier phase, and N_k is a sequence of independent, identically distributed zero-mean Gaussian noise samples with variance σ^2 that models the effects of thermal noise in the RF front end. The signal and noise have been normalized so that the modeled signal amplitude is unity. For convenience, $s_k = c_k \cos(2\pi f_{IF} t_k + \theta_k)$ is used to represent the deterministic signal components. Also for convenience, and without loss of generality, the receiver time t_k is assumed to be equivalent to true time with a uniform sampling interval $T_s = t_k - t_{k-1}$.

The model's security code w_k is a generalization of a binary modulating sequence that is either fully encrypted or contains periodic authentication codes. The defining feature of w_k is that some or all of its symbols are unpredictable to a would-be spoofer prior to broadcast from a legitimate GNSS source. The unpredictable symbols of w_k serve two related functions: they enable verification of w_k as originating from a GNSS Control Segment (standard message authentication) and they make possible a hypothesis test for a security code estimation and replay attack [13]. Various security code implementations will be considered in Sec. III.

B. Attacks against Security-Enhanced GNSS Signals

GNSS spoofing is the transmission of counterfeit GNSS signals with the intent to manipulate the position, velocity, and timing (PVT) readout of a GNSS receiver. A spoofer matches its counterfeit signal structure to that of the authentic signals, as modeled by Eq. (1). To circumvent the security afforded by the unpredictable security code w_k, the spoofer may attempt one of the following specialized spoofing attacks.

B.1 Meaconing

The recording and playback of an entire block of RF spectrum containing an ensemble of GNSS signals [2]. Constituent GNSS signals are not typically separated during record and playback, which implies that a meaconing attack cannot arbitrarily manipulate the PVT of target receivers; rather, target receivers will display the position and velocity of the meaconer and a time in arrears of true time. For a single GNSS signal corresponding to a particular satellite, the combined meaconed and authentic received signals can be modeled as

$$Y_k = \alpha w_{k-d} s_{k-d} + N_{m,k} + w_k s_k + N_k \tag{2}$$

where $N_{m,k}$ is the noise introduced by the meaconer's RF front end, N_k is the noise introduced by the target receiver's RF front end, and $d > 0$ is the number of samples of meaconing delay, such that the meaconed signal $\alpha w_{k-d} s_{k-d}$ arrives at the target receiver with a delay of d samples relative to the authentic signal $w_k s_k$. The coefficient $\alpha > 1$ is the meaconed signal's amplitude advantage factor.

High performance digital signal processing hardware permits a meaconer located close to its intended target to drive the delay d to ever smaller values. In the limit as d approaches zero the attack becomes a zero-delay meaconing attack with the meaconed signals code-phase-aligned with their authentic counterparts. Such alignment enables a seamless liftoff of the target receiver's tracking loops, following which a meaconer can increase d at a rate that is consistent with the target receiver clock drift and gradually impose a significant timing delay.

B.2 Security Code Estimation and Replay (SCER) Attack

Allows greater flexibility than a meaconing attack in manipulating the target receiver's PVT solution. In a SCER attack, a spoofer receives and tracks individual authentic signals and attempts to estimate the values of each signal's unpredictable security code chips on-the-fly. It then reconstitutes a consistent ensemble of GNSS signals, with the security code chip estimates taking the place of the authentic codes, and re-broadcasts these with some delay. For a single GNSS signal corresponding to a particular satellite, the combined SCER-spoofed and authentic received signals can be modeled as

$$Y_k = \alpha \hat{w}_{k-d} s_{k-d} + w_k s_k + N_k \tag{3}$$

where $\hat{w}_{k-d}$ represents the security code estimate arriving with a delay of d samples relative to the authentic security code w_k and other quantities are as described previously. The delay d can be modeled as the sum $d = p + e$ of a processing and transmission delay p, which represents the required signal processing and propagation time and which does not contribute to better estimates of the security code chips, and an estimation and control delay e, which represents an additional delay imposed by the spoofer to improve its estimate of the security code chip values and to control the relative phasing of the spoofed signals so as to impose spoofer-defined position and timing offsets on the target receiver. If the initial delay d exceeds the spreading code chip interval (i.e., if $dT_s > T_c$), then the spoofer will be unable to dislodge the target receiver's tracking loops without forcing re0acquisition. Thus, if the spoofer has an irreducible delay $dT_s > T_c$ then it must first jam or obstruct the incoming GNSS signals to force the target receiver to perform re-acquisition.

The success of a SCER attack depends on the accuracy of the security code estimate. Let k_l be the index of the first sample within the lth authentic security code chip. Then for the received sample Y_{k+d}, with $k_l \leq k < k_{l+1}$, a maximum of $\min(e + k - k_l + 1, \lfloor T_w/T_s \rfloor)$ security code samples will have been summed within the spoofer to produce the security code estimate $\hat{w}_{k+d-d} = \hat{w}_k$, where $\lfloor x \rfloor$ is the floor of x (the largest integer not greater than x). The accuracy of the chip estimates improves with increasing number of participating samples. For example, the probability of error for hard-decision chip estimates is $p_e = \text{erfc}(\sqrt{mT_s(C/N_0)_s})/2$ where m is the number of participating samples at sampling interval T_s, $(C/N_0)_s$ is the spoofer's carrier-to-noise ratio, and $\text{erfc}(\cdot)$ is the complementary error function. Thus, because $m \leq \lfloor T_w/T_s \rfloor$, small T_w severely limits the accuracy of the security code estimates. Consider that a spoofer receiving the legacy Y-code GPS signal, for which $T_w \approx 2$ μs, at a nominal carrier-to-noise ratio of 48 dB-Hz, generates hard-decision chip estimates with a 30% probability of error. A detection strategy for short-delay SCER attacks is detailed in [13].

Long security code chips (e.g., $T_w = 20$ ms for NMA) allow the spoofer to increase e and thereby generate highly accurate chip estimates. A large delay $d = p + e$, however, is itself a liability for the spoofer. The signal denial prelude to a SCER attack must be made long enough that d is consistent with the target receiver's clock drift during the denial interval; otherwise, d will lead to a suspicious increment in the target receiver's pseudorange measurements. Thus, the spoofer finds itself vulnerable to detection at low d due to poor security code chip estimates and at high d due to timing anomalies. This is suggestive of the probabilistic nature of signal authentication, which is further elucidated in the following section.

C. Components of a Comprehensive Probabilistic GNSS Signal Authentication Strategy

In simplest terms, GNSS signal authentication means certifying that a received signal is not counterfeit, that it originates from a GNSS satellite and not a spoofer. As opposed to data authentication, however, GNSS signal authentication is far from absolute; rather, it involves a set of hypothesis tests each with a probability of false alarm. In the formulation adopted here, the tests are designed to detect a spoofing attack under the assumption that a spoofer will either (1) generate a falsified security code that does not match the authentic security code, (2) attempt a non-zero-delay meaconing attack, or (3) attempt a SCER attack. Framed by these assumptions, GNSS signal authentication can be interpreted as involving two authentication sub-types: (1) code origin authentication, a certification that the security code originates with the GNSS Control Segment, and (2) code timing authentication, a certification that the security code arrives promptly and intact.

In the sections that follow, the functional components that support code origin authentication and code timing authentication are described. As a guide to the discussion, the components and their interconnections are presented schematically in Fig. 1 for a security code based on NMA. Adaptations to Fig. 1 for other types of security codes (e.g., GPS Y-code-type encryption or spread spectrum security codes [3]) are discussed further on.

For simplicity of presentation, Fig. 1 represents the authentication process for a single GNSS signal, i.e., a signal identified by a unique combination of spreading code and carrier frequency. An entire ensemble of GNSS signals is assumed to be downmixed and sampled in the RF front end to produce the sampled signal output Y_k, which is routed to the signal tracking and navigation processor where the raw digital output of the RF front end is correlated against receiver-generated signal replicas to acquire and track multiple constituent GNSS signals. However, from the perspective of downstream components, which are associated with a single GNSS signal, Y_k can be modeled as in Eq. (1) for unspoofed signals and in Eqs. (2) and (3) for meaconed and SCER-spoofed signals, respectively.

C.1 Code Origin Authentication

In the case of a security code based on NMA, the signal tracking and navigation processor produces a sequence W_l' of received navigation message symbol estimates. In most cases, these symbols are an error-correction-encoded version of the navigation message data (e.g., the GPS CNAV message is convolutionally encoded before transmission [19]). As the sequence W_l' passes through the error correction decoder, errors introduced by noise in the transmission channel are corrected and the navigation message symbols b_j are recovered. At low carrier-to-noise (C/N_0) ratios some errors may remain in b_j. The code integrity check exploits redundant symbols in b_j (e.g., cyclic redundancy check codes in the GPS CNAV message [19]) to determine whether errors remain. Upon success, the code integrity check sets its logical output I high. For practical purposes, a successful integrity check indicates that the navigation message is correct as received.

The nth block of N_b navigation message symbols $\underline{\mathrm{B}}^n \equiv [b_{j_n}, b_{j_n+1}, ..., b_{j_n+N_b-1}]^T$, which in an NMA scheme includes both navigation data and a digital signature, is passed to a code verification algorithm $\mathbb{V}(\daleth, \underline{\mathrm{B}}^{\ltimes})$ that verifies $\underline{\mathrm{B}}^n$ against a cryptographic key k. If the verification check passes, then $\underline{\mathrm{B}}^n$ can be safely assumed to originate with the GNSS Control Segment. In this case, the logical output signal $H_{1,C}$ remains low. Otherwise, if the verification fails, $H_{1,C}$ is asserted; however this does not necessarily indicate a spoofing attack. Despite error correction, there may yet remain errors in the symbol stream b_j. A single error within the block $\underline{\mathrm{B}}^n$ would cause the code verification to fail. Because of this possibility, and by analogy with other hypothesis tests to be introduced shortly, it is convenient to view the code verification as a statistical hypothesis test. The probability of false alarm for the nth verification is $P_{F,C} = 1-(1-p_{e,j})^{N_b}$, with $p_{e,j}$ being the probability that b_j is wrong, which depends on C/N_0 over the jth symbol, where $j_n \leq j < j_n + N_b$. The output $H_{1,C}$ is combined in a logical 'OR' operation with outputs from other hypothesis tests to produce H_1.

If the code verification fails ($H_{1,C}$ high) but the code integrity check passes (I high), then, with a very high likelihood, the code verification failure cannot be attributed to symbol errors caused by noise. In this case, the output S is asserted, indicating a nearly certain spoofing attack. As opposed to $H_{1,C}$, which goes high with false alarm rate probability $P_{F,C}$ even under normal unspoofed conditions, the infinitesimal probability of false alarm associated with output S suggests that S need not be viewed probabilistically.

One might ask why $H_{1,C}$ should be considered independently from S. The answer is that if only S is considered then a would-be spoofer could always maintain S low by injecting a symbol stream b_j that repeatedly fails the code integrity check. Thus, the outputs S and $H_{1,C}$ are monitored independently both to prevent this type of an attack and in recognition of the clear certainty of a spoofed condition when S goes high.

C.2 Code Timing Authentication

The following functional blocks are involved in code timing authentication: the timing consistency check, the SCER detector, and the jamming-to-noise (J/N) detector.

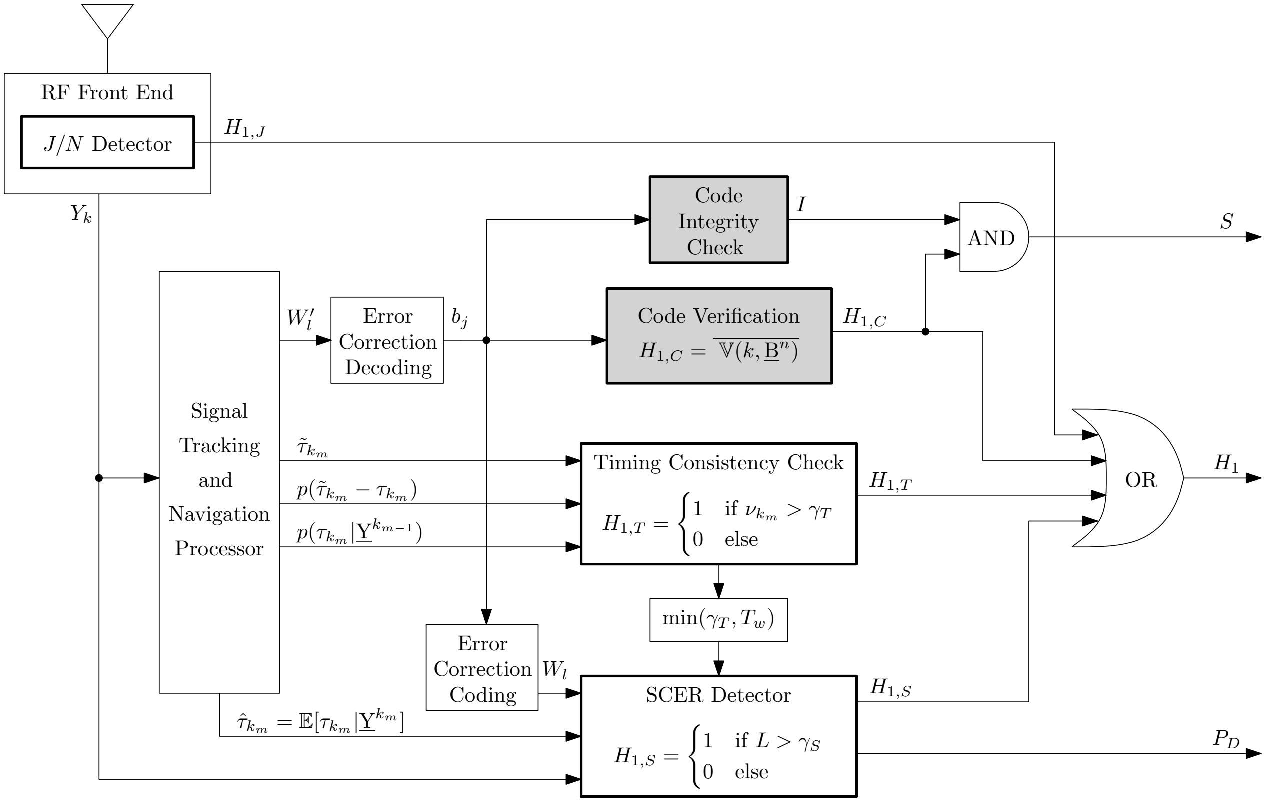
RF Front End
J/N Detector
$H_{1,J}$
Y_k
Signal Tracking and Navigation Processor
W_l'
Error Correction Decoding
b_j
Code Integrity Check
I
AND
S
Code Verification
$H_{1,C} = \overline{\mathbb{V}(k, \underline{\mathrm{B}}^n)}$
$H_{1,C}$
$\tilde{\tau}_{k_m}$
$p(\tilde{\tau}_{k_m} - \tau_{k_m})$
$p(\tau_{k_m}|\underline{\mathrm{Y}}^{k_{m-1}})$
Timing Consistency Check
$H_{1,T} = \begin{cases} 1 & \text{if } \nu_{k_m} > \gamma_T \\ 0 & \text{else} \end{cases}$
$H_{1,T}$
OR
H_1
$\min(\gamma_T, T_w)$
Error Correction Coding
W_l
SCER Detector
$H_{1,S} = \begin{cases} 1 & \text{if } L > \gamma_S \\ 0 & \text{else} \end{cases}$
$H_{1,S}$
$\hat{\tau}_{k_m} = \mathbb{E}[\tau_{k_m}|\underline{\mathrm{Y}}^{k_m}]$
P_D

Fig. 1. Schematic showing GNSS receiver components required for GNSS signal authentication. Components that support code origin authentication are outlined in bold and have a gray fill, whereas components that support code timing authentication are outlined in bold and have no fill. The schematic assumes a security code based on navigation message authentication.

The timing consistency check is a hypothesis test on the timing of the received spreading code c_k. It amounts to a consistency check on the code phase measurement innovation, or the difference between the measured and predicted code phase, and is essentially a special case of so-called receiver autonomous integrity monitoring [20]. The check takes three inputs from the signal tracking and navigation processor:

$\tilde{\tau}_{k_m}$: the receiver's mth measurement of code phase, expressed as the arrival time of some feature of the incoming signal and defined at receiver time t_{k_m}.

$p(\tilde{\tau}_{k_m} - \tau_{k_m})$: the probability distribution of the code phase measurement noise error.

$p(\tau_{k_m}|\underline{\mathrm{Y}}^{k_{m-1}})$: the *a priori* probability distribution of the code phase τ_{k_m} given all input data $\underline{\mathrm{Y}}^{k_{m-1}} \equiv [Y_1, Y_2, ..., Y_{k_{m-1}}]^T$ up to the $(m-1)$th code phase measurement.

In the consistency check, the difference, or innovation, between the measured code phase $\tilde{\tau}_{k_m}$ and the predicted code phase $\bar{\tau}_{k_m} = \mathbb{E}[\tau_{\urcorner_{\gtrdot}}|\underline{\mathrm{Y}}^{\urcorner_{\gtrdot}-\nVdash}]$ is compared against a threshold γ_T. Let $\nu_{k_m} = \tilde{\tau}_{k_m} - \bar{\tau}_{k_m}$ be the innovation. Then the output $H_{1,T}$ is asserted if $\nu_{k_m} > \gamma_T$; otherwise, $H_{1,T}$ remains low. The value of γ_T, which in general varies with time, depends on a pre-selected false alarm probability $P_{F,T}$ for the timing consistency check and on the innovation's conditional distribution, $p(\nu_{k_m}|\underline{\mathrm{Y}}^{k_{m-1}})$, which is derived from $p(\tilde{\tau}_{k_m} - \tau_{k_m})$ and $p(\tau_{k_m}|\underline{\mathrm{Y}}^{k_{m-1}})$. Commonly, the distributions involved can be modeled as Gaussian, in which case $p(\nu_{k_m}|\underline{\mathrm{Y}}^{k_{m-1}})$ can be summarized by its mean $\mathbb{E}[\nu_{\urcorner_{\gtrdot}}|\underline{\mathrm{Y}}^{\urcorner_{\gtrdot}-\nVdash}] = \nvdash$ (assuming an unbiased estimator and unbiased measurements) and variance $\sigma_\nu^2 = \sigma_{\Delta\bar{\tau}}^2 + \sigma_{\Delta\tilde{\tau}}^2$, where $\sigma_{\Delta\bar{\tau}}^2 = \mathbb{E}[(\tau_{\urcorner_{\gtrdot}} - \bar{\tau}_{\urcorner_{\gtrdot}})^{\nvDash}|\underline{\mathrm{Y}}^{\urcorner_{\gtrdot}-\nVdash}]$ and $\sigma_{\Delta\tilde{\tau}}^2 = \mathbb{E}[(\tilde{\tau}_{\urcorner_{\gtrdot}} - \tau_{\urcorner_{\gtrdot}})^{\nvDash}]$. The threshold γ_T is the value of γ for which

$$P_{F,T} = \int_\gamma^\infty p(\nu_{k_m}|\underline{\mathrm{Y}}^{k_{m-1}}) d\nu_{k_m} \tag{4}$$

Note that by comparing ν_{k_m}, not $|\nu_{k_m}|$, against the threshold, the consistency check doubles its sensitivity by making the implicit assumption that the spoofer can only delay the code phase (increase τ_{k_m}).

Another interpretation of γ_T is as the "window of acceptance" referred to in [3]. Between code phase measurement updates, the innovation's conditional distribution $p(\nu_{k_m}|\underline{\mathrm{Y}}^{k_{m-1}})$ widens as receiver clock drift and position uncertainty cause the *a priori* code phase estimate $\bar{\tau}_k$ to become less certain. The distribution can become especially wide if the receiver has a poor clock and is subjected to prolonged jamming or signal blockage. If, after re-acquisition, the innovations remains below γ_T, then the timing of the re-acquired signal is within the window of acceptance; i.e., it is consistent with the assumed uncertainty in $\bar{\tau}_k$.

It should be noted that $p(\nu_{k_m}|\underline{\mathrm{Y}}^{k_{m-1}})$ depends on all signals being tracked by the receiver, not only on the individual signal whose code phase measurement is $\tilde{\tau}_{k_m}$. This is because the *a priori* distribution $p(\tau_{k_m}|\underline{\mathrm{Y}}^{k_{m-1}})$, from which $p(\nu_{k_m}|\underline{\mathrm{Y}}^{k_{m-1}})$ is derived, is a complete summary of what the receiver knows about τ_{k_m} based on all the raw samples in $\underline{\mathrm{Y}}^{k_{m-1}}$. When a particular signal is acquired or re-acquired, its authentication depends on the time aiding provided by other signals. Vector tracking algorithms [21] are particularly well suited for GNSS signal authentication because they combine timing information from all signals and can be designed to produce $p(\nu_{k_m}|\underline{\mathrm{Y}}^{k_{m-1}})$ as part of their routine processing.

The remaining two functional units involved in timing authentication are the J/N detector and the SCER detector. Their operation is summarized only briefly here; a fuller discussion is found in [13] and [22]. The SCER detector is a hypothesis test that decides whether the security code in the incoming samples Y_k arrives (1) intact and (2) near the *a posteriori* code phase estimate $\hat{\tau}_{k_m} = \mathbb{E}[\tau_{\urcorner_{\gtrdot}}|\underline{\mathrm{Y}}^{\urcorner_{\gtrdot}}]$ produced by the signal tracking and navigation processor. At least one of these two conditions is violated if a SCER attack is underway. The SCER detector performs time-weighted correlations with Y_k over the lth unpredictable security chip interval to produce a single-chip statistic S_l. These correlations involve the error correction encoded symbols W_l, which are identical to the raw received symbols W_l' if no symbol errors are present in W_l, but, in general, include corrections to W_l made possible by the operation of error correction decoding and subsequent re-encoding.

The SCER detector combines a set of N single-chip correlations S_l into a detection statistic L, which it compares with a threshold γ_S that is set by a pre-selected probability of false alarm, $P_{F,S}$. If a SCER attack is underway, and if the estimation delay e is sufficiently small, then L will rise above γ_S, causing $H_{1,S}$ to assert. The SCER detector assumes that the spoofer's C/N_0 advantage over the target receiver is limited to approximately 3 dB. It further assumes that a J/N detector is monitoring the incoming in-band power so that the power advantage of the received spoofing signal ensemble is limited to approximately 4 dB above the authentic signal ensemble. This is why a J/N detector is a necessary component of a comprehensive signal authentication strategy. The J/N detector threshold is governed by a pre-determined false alarm probability $P_{F,J}$ [22].

In a typical application, the SCER detector performs a hypothesis test just after each code verification $\mathbb{V}(\mathbb{K}, \underline{\mathrm{B}}^{\ltimes})$. There is little point in performing the test more frequently, since the authenticity of the symbols b_j, and by extension the encoded symbols W_l used in the SCER detector correlations, cannot be guaranteed until the code verification has been performed.

The SCER detector outputs a probability of detection P_D that depends on the detector's model for the statistics of a SCER spoofing attack, which in turn depend on the possible estimation delay e. In setting P_D, the SCER detector pessimistically assumes that the total estimation delay in seconds eT_s could be as large as γ_T, which means that at each security code chip transition the spoofer could already have an estimate based on as much as $\min(\gamma_T, T_w)$ seconds into the upcoming chip. A degraded P_D reflects the penalty paid, in terms of ability to detect spoofing, for uncertainty in ν_{k_m}, which could be caused by an extended period of GNSS jamming or blockage. As $p(\nu_{k_m}|\underline{\mathrm{Y}}^{k_{m-1}})$ widens and γ_T increases, the limitations on spoofing delay d become less stringent. Knowing this, a SCER-attack spoofer can increase the estimation time e, thereby improving the reliability of its security code chip estimates. When the spoofer's $(C/N_0)_s$ is high and γ_T is large (e.g, $(C/N_0)_s > 50$ dB-Hz and $\gamma_T > 300$ μs), then the null and spoof hypotheses become virtually indistinguishable within the SCER detector and P_D drops. Even though γ_T may subsequently contract and P_D increase, a low P_D creates a window of vulnerability after which signal authentication assurance is permanently degraded.

C.3 Other Security Code Implementations

The above components of a GNSS signal authentication system are specific to a security code based on NMA. The components are also valid for the civil public spreading code authentication technique introduced in [3] except that in this case the symbols b_j are routed directly to the SCER detector where they are used to seed a pseudorandom spreading code generator a segment of whose output gets inserted into the local spreading code replica.

For private spreading code authentication schemes such as the civil level-3 technique introduced in [3] and military GPS Y- and M-code security, the code verification block in Fig. 1 is unnecessary. The figure can be adapted to these cases by setting $H_{1,C}$ permanently low and by routing the symbols b_j directly to the SCER detector. These private-key techniques rely on storage of a secure "red key" in tamper-resistant hardware within the receiver. Segments of the symbol stream b_j are coupled with the red key in the SCER detector to produce a seed for a pseudorandom spreading code generator. Only segments of the generated code are used in the civil private-key technique of [3], whereas the continuous output of the generator constitutes the security code for GPS Y- and M-code security.

D. Operational Definition of GNSS Signal Authentication

With the authentication components and their interactions specified, an operational definition of GNSS signal authentication—in other words, how signals are declared authentic in practice—can now be formulated. A GNSS signal is declared authentic at a given moment if and only if, during the time elapsed since some initialization event at which the receiver was known to be tracking only genuine GNSS signals, (1) the logical output S has remained low, (2) the logical output H_1 has remained low, and (3) the real-valued output P_D has remained above an acceptable threshold (e.g., 0.9).

Some comments about this operational definition are in order. First, although there may be reasonable alternatives to this definition, they cannot be substantially different. Aside from the variations discussed in Sec. II-C.3, the components of the proposed definition are each unique and necessary. Second, although a GNSS signal may be pronounced authentic by the above operational definition, it may in fact be counterfeit. Practical constraints of hypothesis testing prevent P_D from reaching unity. For example, for the NMA-based security codes discussed later on, nominal P_D may drop as low as 0.97. Moreover, jamming or signal blockage can temporarily reduce P_D. Inversely, even though a signal may be declared unauthentic, it may actually be authentic. In the case that S is asserted, the incoming signal is certainly unauthentic; on the other hand, H_1 will at times assert even under unspoofed conditions. It has a false alarm probability

$$P_F = 1 - (1 - P_{F,J})(1 - P_{F,C})(1 - P_{F,T})(1 - P_{F,S})$$

which is greater than any of the false alarm probabilities for the individual tests that can trigger H_1. Third, movement of P_D below the acceptable threshold does not necessarily indicate a SCER spoofing attack, it only indicates that the SCER detector's probability of detecting a SCER attack has been compromised, and thus the currently tracked signal cannot be considered authentic.

E. Remarks

It is easy to appreciate the advantage of short over long security code chips given the authentication architecture proposed in Fig. 1. Short chips such as the $T_w \approx 2$ μs chip of the legacy GPS Y code keep $\min(\gamma_T, T_w)$ to less than a few microseconds and thereby prevent significant degradation in P_D even during a prolonged signal blackout, whereas long chips such as $T_w \approx 20$ ms for NMA allow significant degradation in P_D for the same outage. This weakness of NMA-based GNSS signal authentication has been noted—although not in these formal terms—in [3] and [7]. Practically, the weakness translates into the following additional requirements for NMA-based GNSS security: For a static receiver in a known location, maintaining P_D high requires either continuous tracking of at least one strong GNSS signal or a clock that does not drift significantly during whatever complete signal outages occur. For a receiver mounted on a dynamic platform, either continuous tracking of at least 4 strong GNSS signals or a clock and inertial measurement unit (IMU) combination that does not drift significantly are required.

Given these requirements, one may question whether NMA-based GNSS security will be useful in practice. One should bear in mind that for many applications of interest the prolonged signal denial required to significantly degrade P_D would be highly suspicious. For example, consider a static receiver with a low-cost temperature-compensated crystal oscillator having 10^{-8} stability. A spoofer would be forced to preface a spoofing attack with a 150-second complete signal denial interval in order to increase γ_T beyond 5 μs (assuming $P_{F,T} < 0.002$) and thereby cause a significant reduction in P_D [13]. If the complete signal denial is done via jamming, then the J/N detector will trigger; if done by obstructing the target receiver's antenna, this requires close physical access. In any case, the signal outage will appear suspicious.

Also, it is worth noting that security code alternatives to NMA are not foolproof and are likely to be less practical. Indeed, it appears that no exclusively cryptographic defense, no matter how short the security chip interval T_w, can detect a well-executed near-zero-delay meaconing attack. (This is why such an attack is excluded from Section II-C's attack model.) Universal vulnerability to near-zero meaconing suggests the need for a layered approach that combines cryptographic signal authentication with non-cryptographic techniques such as the vestigial signal defense [23]. It also suggests that expectations for GNSS signal authentication must be modest: the goal should not be preventing a successful attack at all cost, but making one difficult. Furthermore, a GNSS signal authentication scheme's potency must be weighed against its practicality. This tradeoff is the subject of the next section.

III. PROPOSED GPS CRYPTOGRAPHIC SIGNAL AUTHENTICATION STRATEGY

The previous section considered general GNSS signal authentication, which relies in part on some or all of the security code w_k being unpredictable to a would-be spoofer. This section considers a navigation message authentication (NMA) strategy specifically for civil GPS because it is (1) *effective*—NMA makes it more difficulty for a spoofer to carry off a successful spoofing attack—and (2) *practical*—it is likely to be adopted by the GPS community. For full explanation of the following discussion, see Sections III and IV of Ref. [12].

Navigation message authentication is a scheme that sets

$w_k = d_k$ where d_k are samples from the ±1-valued navigation message and $T_w = 20$ ms. One can either make all or part of the navigation message unpredictable to generate w_k. A practical strategy is to form w_k by introducing periodic randomness into the navigation message. This NMA-based approach is assumed hereafter. The periodic randomness is best introduced by public key cryptographic digital signatures. Public key digital signatures are practical because they can store the encryption keys in unencrypted receiver memory (i.e., a receiver need not have tamper-proof hardware [24]).

The Elliptic Curve Digital Signature Algorithm (ECDSA) is a public key digital signature standard operates on groups associated with an elliptic curve space [18]. Among other reasons, ECDSA is recommended for NMA because ECDSA has efficient verification algorithms [25–27], and NSA recommends that systems built after 2010 implement ECDSA [28]. The standardized ECDSA 233-bit Koblitz curve (K-233) is attractive for civil GPS signal authentication because it generates a short 466-bit signature amenable to optimized software-defined verification routines [28, 29]. To sign messages, ECDSA first applies a secure hash function to generate a digital fingerprint of the message, which is typically shorter than the message itself, and then signs the fingerprint rather than the whole message. In selecting the appropriate hash function for GPS signal authentication, NIST offers a standardized cryptographic hash family named SHA-2 [30]. Since there is no computational difference between SHA-224 and the stronger SHA-256, SHA-256 is proposed for implementation.

This remainder of the section offers a concrete strategy for cryptographic civil GPS signal authentication. The strategy is based on public key elliptic curve cryptographic signatures inserted periodically into the flexible GPS civil navigation (CNAV) message. Specific details of the strategy, offered here, facilitate near-term adoption by the GPS Control Segment. The proposed strategy enables civil GPS signal authentication as described in Sec. II and diagrammed in Fig. 1 with the following properties: (1) a probability of detection of $P_D > 0.97$ for $P_{F,S} = 0.0001$, (2) a cryptographic strength (i.e., symmetric key equivalent strength) of 112 bits, and (3) authentication every five minutes per channel.

A. Digital Signature Conveyance via CNAV

The flexible CNAV message format that modulates modernized GPS signals offers a convenient conveyance for a digital signature. The CNAV format was designed to be extensible so that new messages can be defined within the framework of the GPS IS. The CNAV message format is broadcast from Block IIR-M GPS spacecraft at the L2 frequency and Block IIF GPS spacecraft at the L2 and L5 frequencies [19]. Plans call for CNAV to be broadcast from Block IIIA GPS spacecraft at the L2 and L5 frequencies and additionally at L1. Thus, future single-frequency receivers can benefit from the extension to the CNAV message proposed in this section.

Every 12 seconds, a CNAV message delivers a 300-bit packet, which includes a 38-bit header, a 238-bit payload, and a 24-bit cyclic redundancy check (CRC). The flexibility of CNAV is due in part to the information broadcast over the header, which delivers a 6-bit message type identification field identifying up to 64 unique message types. The current GPS IS defines only 15 of these messages, reserving the others for future applications [19].

The following proposal defines two new CNAV messages to deliver an ECDSA signature. This is not a fundamental change to the GPS IS, but rather an extension to CNAV. Thus, this extension to CNAV can be considered practical in the sense defined earlier.

B. CNAV Message Signature Type Definition

Since the CNAV structure does not support payloads larger than 238 bits, the 466-bit ECDSA signature selected at the end of the last section must be broadcast across two CNAV messages. It is proposed to define two CNAV messages that deliver the 466-bit ECDSA signature, each message having the format shown in Fig. 2. The first ECDSA CNAV message type contains the first 233-bit half of the signature and the second message type contains the second half of the signature.

A 466-bit signature broadcast over two 238-bit payloads leaves 10 bits undefined. It is proposed to uniquely and randomly generate these bits for each instance of a signed message with a standardized pseudorandom number generator [31]. This technique is known as adding cryptographic "salt." Since the 10 salt bits are unpredictable prior to broadcast, they contribute to the total number of unpredictable w_k symbols available to a receiver to perform SCER detection tests. However, they do not increase the signature's strength since they are not part of the digital signature. Like other components of the navigation message, they are digitally signed and can therefore be authenticated as originating from the Control Segment. Together, the two CNAV signature messages transmit 476 unpredictable bits.

C. Signing the CNAV Message

The frequency at which the CNAV navigation message can broadcast signatures requires consideration of several factors. First, although the CNAV message format is flexible, it is not without constraints. Ephemeris message types 10 and 11 and a timing message of type 30–39 must be broadcast at least every 48 seconds to ensure accurate GPS receiver operation [5,19]. Since a practical signal authentication strategy cannot adversely affect a receiver's position

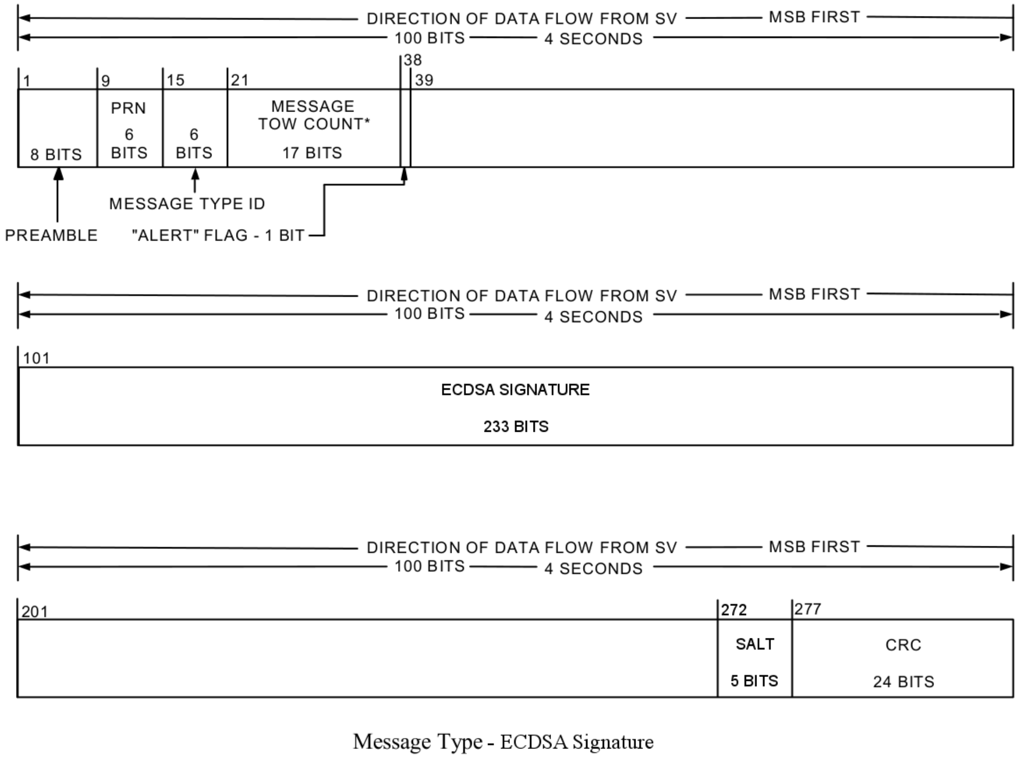

Fig. 2. Diagram showing the format of the proposed CNAV ECDSA signature message, which delivers the first or second half of the 466-bit ECDSA signature and a 5-bit salt in the 238-bit payload field (figure adapted from [19]).

solution, the CNAV signature must respect these requirements. Given these constraints, the smallest block of data in which a complete signature can be embedded is the 96-second signature block such as the one shown in Fig. 3. In this structure, the two CNAV signature messages are interleaved between the ephemeris and clock data to meet the broadcast requirements.

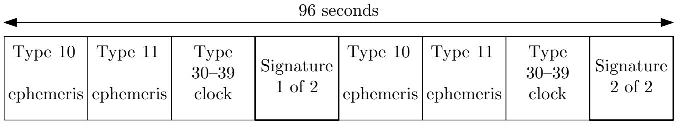

Fig. 3. Schematic illustrating the shortest broadcast signature block that does not violate the CNAV ephemeris and timing broadcast requirements. To meet the required broadcast interval of 48 seconds for message types 10, 11, and one of 30–39, the ECDSA signature is broadcast over a 96-second signature block that is composed of eight CNAV messages.

A second consideration when signing the CNAV message is the duration between signature blocks. This choice involves a tradeoff between effectiveness (i.e., offering frequent authentication) and practicality (i.e., imposing a low computational burden relative to standard GPS signal processing and maintaining a low percentage of the CNAV message reserved for the digital signature). The maximum rate at which the CNAV message can be signed corresponds to a scenario in which the 96-second signature block in Fig. 3 is broadcast continuously back-to-back. However, this strategy is not practical: besides the high percentage of the navigation message reserved for the signature (i.e., 25%), this back-to-back configuration would eliminate the possibility of sending any other message types than 10, 11, 30–39, and the signature. Instead, a reasonable approach would be to sign every 336 seconds (about every five minutes). In this case, one signature block would authenticate every 28 CNAV messages as illustrated in Fig. 4. This means the percentage of the navigation message devoted to the digital signature is a more practical 7.5%.

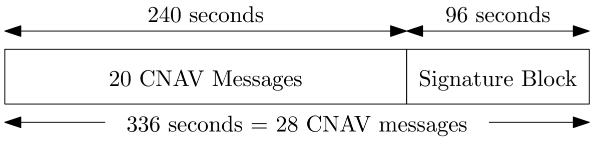

Fig. 4. Schematic illustrating a signed 336 second broadcast. The proposed strategy signs every 28 CNAV messages with a signature broadcast over two CNAV messages on each broadcast channel.

To broadcast a signature every five minutes, the Control Segment would first compute the next five minutes worth of CNAV navigation message including the salt. It would then concatenate signable navigation message bits in order—that is the first 23 CNAV messages (i.e., the 20

CNAV message in Fig. 4 and the first three in Fig. 3), the first signature header, the first five bits of the salt, the 5th through 7th CNAV messages from Fig. 3, the second signature header, and remaining five salt bits—and then generate the SHA-256 fingerprint. After generating the ECDSA signature from the fingerprint, the Control Segment would break the signature into two parts and insert each part into a ECDSA signature message shown in Fig. 2. These two signature messages would then be transmitted at the appropriate times as part of the CNAV message signature block as seen in Fig. 3.

Note that the signature and corresponding CRC are not themselves signed. This is because neither is known until after signature generation. Unlike the signature field, which is entirely unpredictable, the CRC can be deterministically computed by a receiver immediately upon receiving the last unpredictable bit of any CNAV message. Thus, the CRC symbols cannot be used for SCER detection.

It is worth noting that a single bit error would cause the verification algorithm to fail. CNAV has the option of being broadcast with forward error correction enabled. As described in Sec. II, FEC would enhance the robustness of NMA-based signal authentication. It is therefore recommended that FEC be enabled to support civil GPS signal authentication.

D. Constellation-Wide Signature Scheduling

Under the proposed strategy, each channel is authenticated every five minutes. However, the per-channel signature block could be offset from other channels (i.e., other satellites in the GPS constellation) such that a receiver tracking several satellites would see signatures more frequently. This offset strategy would substantially constrain the degrees-of-freedom that a spoofer could manipulate. An optimal offset strategy would minimize the maximum time between authentications T_{ba} [i.e., $\min(\max(T_{ba}))$] that a receiver at any point on earth between a certain upper and lower latitude would observe based on the current constellation spatial arrangement. The optimal satellite offset assignment problem can be reduced to a directional graph coloring problem [32] that is likely best solved via a genetic algorithm similar to the one proposed for use in future optimization of the GPS constellation itself [33]. A sub-optimal solution computed through a greedy algorithm for the constellation in August 2011 computed that $\min(\max(T_{ba})) = 144$ seconds was possible between $\pm 70^\circ$ latitude. Thus, even with a simple sub-optimal signature offset assignment, a receiver could receive signatures with a T_{ba} of at most about two minutes and a T_{ba} on average of about one minute.

E. Implementation Details

The receiver modifications required to exploit the proposed civil GPS signal authentication strategy can be readily implemented on a software-defined receiver such as those presented in [34, 35] and [36]. A traditional receiver with application-specific correlation hardware would require some redesign to take advantage of the proposal. First, the correlation hardware would need to be modified to accommodate the new correlations needed for SCER detection [13]. Second, a traditional receiver would need to monitor J/N. This could be a natural extension of the GNSS spectrum monitoring that some GNSS receivers already offer [37, 38]. Third, the traditional receiver would need to implement the remaining elements of Fig. 1 such as signature verification and the timing consistency check in its baseband processor, which is typically a general-purpose processor that is modifiable via firmware updates.

For more implementation details, see Ref. [12].

IV. AUTHENTICATION PERFORMANCE

The proposed civil GPS signal authentication strategy broadcasts 476 unpredictable symbols approximately every five minutes. Given this, the P_D output in Fig. 1 can now be computed for a given threat model based on the statistical tests in [13]. To appreciate the effectiveness of the proposed authentication strategy, consider the following challenging scenario from the target receiver's perspective:

- the spoofer has a 3 dB carrier-to-noise ratio advantage over the receiver (i.e., $(C/N_0)_s = (C/N_0)_r + 3$ dB);
- the received spoofed signals are 1.1 times stronger than the received authentic signals;
- the spoofer has introduced a timing error of 1 μs in the receiver through jamming or other means and exploits this entire delay to improve its estimates of the security code chip values (i.e., the quantity e from Sec. II-B.2 is equal to 1 μs); and,
- the false alarm probability for the SCER detector in Fig. 1 is $P_{F,S} = 0.0001$.

The statistics developed in [13] can be used to show that, under this scenario, the output P_D in Fig. 1 will be maintained above 0.97 over a wide range of authentic signal carrier-to-noise ratio $(C/N_0)_r$ values as seen in Fig. 5. This indicates that the proposed NMA-based strategy enables effective anti-spoofing.

V. CONCLUSION

This paper refines the meaning of GPS signal authentication and offers a practical technique to authenticate civil GPS signals. The proposed technique embeds digital signatures in the GPS civil navigation (CNAV) message and exploits a recently-developed statistical hypothesis test to secure civil GPS receivers against replay-type spoofing attacks. In a challenging example scenario, the technique was shown to detect a replay-type spoofing attack with probability of detection greater than 0.97 for a false alarm probability of 0.0001. The proposed strategy enables receivers to authenticate each individual civil GPS signal

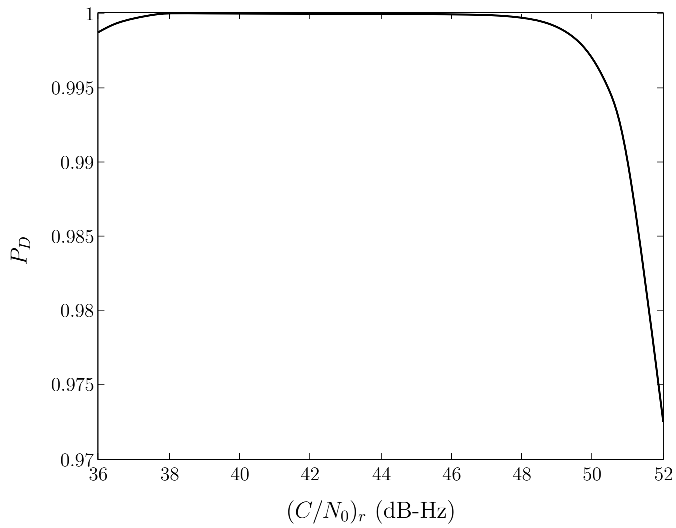

Fig. 5. P_D as a function of $(C/N_0)_r$ for a challenging spoofing attack scenario. The proposed civil GPS signal authentication strategy maintains $P_D > 0.97$ for $P_{F,S} = 0.0001$ over the range of $(C/N_0)_r$ shown.

every five minutes.

ACKNOWLEDGMENTS

The authors thank the members of the University of Texas at Austin Radionavigation Laboratory. A portion of this work was supported by the Department of Defense (DoD) through the National Defense Science and Engineering Graduate Fellowship (NDSEG). Work on this paper was also supported in part by the U.S. Air Force under the STTR project titled “Connected Autonomous Space Environment Sensors (CASES).”

References

[1] W. Clinton, “Statement by the president regarding the United States decision to stop degrading Global Positioning System accuracy,” *Office the the Press Secretary, The White House.*

[2] Anon., “Vulnerability assessment of the transportation infrastructure relying on the Global Positioning System,” John A. Volpe National Transportation Systems Center, Tech. Rep., 2001.

[3] L. Scott, “Anti-spoofing and authenticated signal architectures for civil navigation systems,” in *Proceedings of the ION GNSS Meeting.* Portland, Oregon: Institute of Navigation, 2003, pp. 1542–1552.

[4] M. Kuhn, “An asymmetric security mechanism for navigation signals,” in *Proc. of the 6th Int. Information Hiding Workshop.* Springer, May 2004, pp. 239–252.

[5] C. Wullems, O. Pozzobon, and K. Kubik, “Signal authentication and integrity schemes for next generation global navigation satellite systems,” in *Proc. European Navigation Conference GNSS*, Munich, July 2005.

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

[7] ——, “Authenticating GNSS: Proofs against spoofs, Part 2,” *Inside GNSS*, pp. 71–78, September/October 2007.

[8] P. Papadimitratos and A. Jovanovic, “Protection and fundamental vulnerability of GNSS,” in *IEEE Int. Workshop on Satellite and Space Communications*, 2008, pp. 167–171.

[9] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O’Hanlon, and P. M. Kintner, Jr., “Assessing the spoofing threat: development of a portable GPS civilian spoofer,” in *Proceedings of the ION GNSS Meeting.* Savannah, GA: Institute of Navigation, 2008.

[10] P. Y. Montgomery, T. E. Humphreys, and B. M. Ledvina, “A multi-antenna defense: Receiver-autonomous GPS spoofing detection,” *Inside GNSS*, vol. 4, no. 2, pp. 40–46, April 2009.

[11] O. Pozzobon, “Keeping the spoofs out: Signal authentication services for future GNSS,” *Inside GNSS*, vol. 6, no. 3, pp. 48–55, May/June 2011.

[12] K. Wesson, M. Rothlisberger, and T. E. Humphreys, “Practical cryptographic civil GPS signal authentication,” *NAVIGATION, Journal of the Institute of Navigation*, 2011, submitted for review; available at http://radionavlab.ae.utexas.edu/nma.

[13] T. E. Humphreys, “Detection strategy for cryptographic GNSS anti-spoofing,” *IEEE Transactions on Aerospace and Electronic Systems*, 2011, submitted for review; available at http://radionavlab.ae.utexas.edu/detstrat.

[14] T. Stansell, “Location assurance commentary,” *GPS World*, vol. 18, no. 7, p. 19, 2007.

[15] S. C. Lo and P. K. Enge, “Authenticating aviation augmentation system broadcasts,” in *Proceedings of the IEEE/ION PLANS Meeting.* Palm Springs, California: Institute of Navigation, 2010, pp. 708–717.

[16] O. Pozzobon, C. Wullems, and K. Kubic, “Secure tracking using trusted GNSS receivers and Galileo authentication services,” *Journal of Global Positioning Systems*, vol. 3, no. 1-2, pp. 200–207, 2004.

[17] M. Tran, Y. Lee, J. L. Davis, Jr., J. Rushanan, and C. Hegarty, “An assessment of the applicability of GPS anti-jam and anti-spoof technologies to civil aviation,” The MITRE Corporation, Tech. Rep., 2002.

[18] C. Paar and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners.* Springer, 2010.

[19] Anon., “IS-GPS-200E: Navstar GPS space segment/navigation user interfaces,” Science Applications International Corporation, Tech. Rep., 2010, http://www.losangeles.af.mil/library/factsheets/factsheet.asp?id=9364.

[20] R. G. Brown, *Global Positioning System: Theory and Applications.* Washington, D.C.: American Institute of Aeronautics and Astronautics, 1996, vol. 2, ch. 5: Receiver Autonomous Integrity Monitoring, pp. 143–168.

[21] M. Lashley, D. Bevly, and J. Hung, “Performance analysis of vector tracking algorithms for weak GPS signals in high dynamics,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 3, no. 4, pp. 661–673, 2009.

[22] T. E. Humphreys, D. Shepard, and J. Bhatti, “A testbed for developing and evaluating GNSS signal authentication techniques,” *IEEE Transactions on Aerospace and Electronic Systems*, 2011, in preparation; available at http://radionavlab.ae.utexas.edu/testbed.

[23] K. Wesson, D. Shepard, J. Bhatti, and T. E. Humphreys, “An evaluation of the vestigial signal defense for civil GPS anti-spoofing,” in *Proceedings of the ION GNSS Meeting.* Portland, Oregon: Institute of Navigation, 2011.

[24] O. Pozzobon, C. Wullems, and M. Detratti, “Tamper resistance: Security considerations for GNSS receivers,” *GPS World*, pp. 37–41, April 2011, to appear.

[25] K. Lauter, “The advantages of elliptic curve cryptography for wireless security,” *IEEE Wireless Communications*, vol. 11, no. 1, pp. 62–67, 2004.

[26] D. Hankerson, S. Vanstone, and A. Menezes, *Guide to elliptic curve cryptography.* Springer-Verlag, 2004.

[27] A. Koblitz, N. Koblitz, and A. Menezes, “Elliptic curve cryptography: the serpentine course of a paradigm shift,” *Journal of Number Theory*, 2009.

[28] Anon., “Digital signature standard,” National Institute of Standards and Technology, FIPS PUB 186-3, June 2009.

[29] J. A. Solinas, “Efficient arithmetic on Koblitz curves,” *Designs, Codes, and Cryptography*, pp. 195–249, 2000.

[30] Anon., “Secure hash standard,” National Institute of Standards and Technology, FIPS PUB 180-3, October 2008.

[31] E. Barker and J. Kelsey, “Recommendation for random number generation using deterministic random bit generators (revised),” National Institute of Standards and Technology, NIST Special Publication 800-90, March 2007.

[32] T. R. Jensen and B. Toft, *Graph Coloring Problems*, ser. Wi-

ley Series in Discrete Mathematics and Optimization. Wiley-Interscience, 1994.

[33] E. Lansard, E. Frayssinhes, and J.-L. Palmade, "Global design of satellite constellations: a multi-criteria performance comparison of classical Walker patterns and new design patterns," *Acta Astronautica*, vol. 42, no. 9, pp. 555–564, 1998.

[34] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, and P. M. Kintner, Jr., "GNSS receiver implementation on a DSP: Status, challenges, and prospects," in *Proceedings of the ION GNSS Meeting*. Fort Worth, TX: Institute of Navigation, 2006.

[35] T. E. Humphreys, J. Bhatti, T. Pany, B. Ledvina, and B. O'Hanlon, "Exploiting multicore technology in software-defined GNSS receivers," in *Proceedings of the ION GNSS Meeting*. Savannah, GA: Institute of Navigation, 2009.

[36] B. O'Hanlon, M. Psiaki, S. Powell, J. Bhatti, T. E. Humphreys, G. Crowley, and G. Bust, "CASES: A smart, compact GPS software receiver for space weather monitoring," in *Proceedings of the ION GNSS Meeting*. Portland, Oregon: Institute of Navigation, 2011.

[37] C. Fenger, "u-blox 6 GPS receivers enhanced with many new features," ublox, Tech. Rep. GPS-X-11012, July 2011.

[38] *TRIUMPH-VS Datasheet*, Rev. 2.6 ed., Javad, June 2011.