A Blueprint for Civil GPS Navigation Message Authentication

Andrew J. Kerns, Kyle D. Wesson
Department of Electrical and Computer Engineering
The University of Texas at Austin
akerns@utexas.edu, kyle.wesson@utexas.edu

Todd E. Humphreys
Department of Aerospace Engineering
The University of Texas at Austin
todd.humphreys@mail.utexas.edu

***Abstract*—A proposal for civil GPS navigation message authentication (NMA) is presented with sufficient specificity to enable near-term implementation. Although previous work established the practicality and efficacy of NMA for civil GPS signal authentication, there remains a need for a detailed proposal that addresses several outstanding considerations regarding implementation. In particular, this paper (1) provides a definitive evaluation of the tradeoffs involved in the choice of cryptographic protocol, and (2) optimizes the placement of digital signature bits in the GPS CNAV message stream. By offering GPS engineers and policymakers a detailed blueprint for civil NMA, this work advances the possibility of NMA implementation on modernized civil GPS signals.**

I. Introduction

Manipulation of a GPS receiver's navigation and timing solution by transmitting counterfeit civil GPS signals has been demonstrated using low-cost commercial equipment [1]–[3]. GPS spoofing attacks exploit the transparency and predictability of civil GPS signals. Civil GPS waveforms are defined in public documents and the navigation data modulated onto the waveforms are highly predictable [4]. Thus counterfeit civil GPS signals are easy to generate and transmit.

Navigation message authentication (NMA) is a proposed civil GNSS signal authentication technique in which controlled cryptographic signatures sign navigation data broadcast by GNSS satellites. Besides NMA, a number of promising methods are currently being developed to defend against civil GNSS spoofing attacks [5]–[10]. Nonetheless, NMA is unique in imposing minimal burden on a low-cost receiver: no hardware modifications are required and the increase in processing demands is negligible.

Previous work has shown that NMA, when properly implemented at the transmitter and receiver, would vastly improve civil GNSS security [11]–[15]. It is clear that NMA-enabled receivers can confirm that navigation data are authentic, thereby performing *data authentication*. More surprisingly, previous work demonstrated that, by modulating unpredictable yet verifiable data onto GNSS signals, NMA allows a receiver to detect a broad category of GNSS spoofing attacks known as security code estimation and replay (SCER) attacks [15]. *Signal authentication* confirms that the underlying GNSS signal is authentic, according to an operational definition in [14]. NMA-enabled receivers can implement a SCER detector and other elements of a signal authentication procedure, provided that the receiver possesses a sufficiently accurate time estimate (e.g., μs-level) [14].

A receiver that implements these signal and data authentication techniques, while tracking NMA-enhanced civil GNSS signals, can securely estimate its position, velocity, and time.

While the benefits of NMA are convincing, its implementation does require changes to the GNSS signal structure. These changes are achievable in modernized GPS via the flexible and extensible civil navigation (CNAV) messaging format. Reference [14] argues that the most practical and effective signature generation method for NMA is a public-key authentication protocol known as the Elliptic Curve Digital Signature Algorithm (ECDSA). An alternative to ECDSA, Timed Efficient Stream Loss-Tolerant Authentication (TESLA), is recommended in studies on NMA for space-based augmentation systems and eLORAN [12], [13]. In this paper, ECDSA and TESLA are compared for purposes of civil GPS NMA, and a hybrid scheme is developed to exploit the advantages of the two methods.

Further, this paper studies the allocation of reserved and available CNAV bits for periodic signatures. By making the optimistic assumption that reserved bits in existing CNAV messages are available for transmitting signatures, the proposal demonstrates how to compactly insert NMA signatures into CNAV and analyzes the tradeoff between signature overhead and authentication frequency. Definitions for new CNAV message types are proposed. The questions resolved in this paper clarify the implementation details of NMA and enhance NMA's stature as a practical solution for civil GPS signal authentication.

II. Selection of Cryptographic Methods

The implementation of NMA requires the selection of an algorithm for generating and verifying cryptographic digital signatures. Low-overhead broadcast authentication techniques generally rely on symmetric-key cryptography [16]. Appending signed data with a message authentication code (MAC) allows a recipient to authenticate the data, provided that both the sender and recipient know a shared secret key. Critically, MACs can be much shorter than equivalently-secure signatures generated by asymmetric algorithms, such as ECDSA. However, a key property for civil GPS NMA is asymmetry—only the GPS Control Segment can sign messages. Thus, civil

GPS NMA requires a cryptographic method with asymmetric properties. Two such techniques—ECDSA and TESLA—have emerged in literature as candidate solutions [12]–[14]. According to the U.S. National Institute of Standards and Technology (NIST) guidelines, cryptographic authentication methods are secure beyond 2030 only if their equivalent symmetric-key strength b_s is at least 128 bits [17]. To adhere to NIST recommendations without incurring excess overhead, all methods proposed in this paper will have strength $b_s = 128$ bits.

A. ECDSA

ECDSA is a NIST-standardized digital signature scheme where the signing party (e.g., GPS Control Segment) uses a private key to generate a signature that can be verified using the corresponding public key. If periodic digital signatures are inserted in the broadcast navigation data, an NMA-capable receiver can authenticate the received signal upon reception of both the signed data and the signature [14]. An authentication event (AE) occurs at the earliest time that the verification routine can complete. ECDSA-based NMA is vulnerable to navigation data replay attacks, in which an attacker retransmits old CNAV messages, if the user does not have time knowledge. However, ECDSA-based NMA is not vulnerable to navigation data spoofing attacks, in which an attacker produces an alternate data and signature pair that passes verification tests. Assuming $b_s = 128$ bits and use of an elliptic curve over a prime field (e.g., NIST's P-256 curve), ECDSA public keys and signatures are 256 and 512 bits in length, respectively [18]. Although alternative standardized curves are available, some of which have longer keys and signatures, this paper assumes 512-bit signatures [18], [19]. Thus, the overhead requirement for each AE is 512 bits.

B. TESLA

TESLA is a broadcast authentication scheme which combines symmetric cryptographic primitives with the assumption of loose time synchronization to achieve asymmetric properties [20]. The signing party, henceforth assumed to be the GPS Control Segment, generates a one-way chain of keys and disperses the last key in the chain as if it were a public key. The TESLA transmission algorithm progresses in a reverse direction along the key chain, using a specific key to compute a MAC and then revealing that key after a delay δ. Users receive signed data and an authentication code and, after waiting for the disclosure delay δ, receive the plaintext key. To verify received data, users (1) apply the prescribed one-way function to the new key to check that it matches a previously-established key, thereby demonstrating the authenticity of the new key, and (2) compute a MAC using the data and key, just as the Control Segment did, and check that it matches the broadcast MAC. This paper assumes a specific variant of TESLA in which each key is only used to compute a single MAC. An AE occurs when a user receives and verifies both components of the MAC-key pair.

C. TESLA Key Schedule

The delay δ is critical: before the delay, the key is secret and is thus useful for demonstrating data authenticity, but after the delay, the key is public. As such, δ must be appreciable. For example, $\delta = 880$ ms is proposed later in the paper.

Consider a schedule that defines an exact disclosure time for each element of the TESLA key chain. Let δ_s be the difference between this schedule and when a GPS satellite actually broadcasts a key, and let δ_r be the error in a user's time estimate. For safe authentication, the condition $|\delta_s| + |\delta_r| < \delta$ must hold. For a user to verify if his timing accuracy is sufficient, the Control Segment must publish limits on δ_s. To offer any improvement in operational flexibility, δ_s must be at least 12 seconds (i.e., one CNAV message), which far exceeds the values of δ proposed in later sections. Thus, the best strategy is to broadcast keys at the exact times specified by the key schedule so that δ_s is equal to the satellite clock error. As the satellite clock error is negligible when compared to δ_r and δ, let $\delta_s = 0$. In effect, a reasonably small δ forces the Control Segment to guarantee specific broadcast slots for key-carrying NMA messages, removing any flexibility in scheduling those messages.

D. TESLA MAC Truncation

When analysis depends on the choice of MAC construction, this paper assumes the keyed-hash MAC (HMAC) construction. When using the HMAC construction, the MAC length is the output length of the chosen hash function [21]. For example, SHA-256 results in 256-bit MACs. However, it is common practice to truncate a MAC to form a MAC tag consisting of the m left-most bits of the MAC. To understand the security implications of a MAC tag length m, consider two types of attacks against TESLA: key recovery attacks and MAC tag forgery attacks.

a) Key recovery: A key recovery attack discovers a future element of the key chain, or an alternate key that, once the one-way function is applied, matches a previously-disclosed key. Since the key derivation method uses a suitable cryptographic hash function, this search is computationally infeasible, with 2^{128} complexity. Note that decreasing m does not aid such a search [21], [22]. A successful key recovery attack yields control over the victim receiver until the disclosure time of the discovered key. If over-the-air re-keying is used, and the discovered key is far enough along the key chain (e.g., the root key), control is indefinite.

b) MAC tag forgery: To perform MAC tag forgery, an attacker transmits an invalid message or MAC tag without knowing if the MAC tag will pass the victim receiver's verification test. Since MACs (and thus MAC tags) appear as random values to parties without access to the secret key K, the probability of successfully forging a specific MAC tag is 2^{-m}. If the attacker makes n forgery attempts against one user, the probability of at least one successful attack increases to $n2^{-m}$. Since MAC tags are transmitted according to a regular schedule, an attacker is limited in the number of forgery attempts. For example, if $m = 32$, and a MAC

tag is transmitted every 144 seconds, the probability of a continuous attack succeeding within 10 years is about 1 in 2,000. Unlike key recovery, successful MAC forgery attacks afford an attacker minimal and temporary benefit. Thus, to dutifully minimize authentication overhead, one should choose a small m, but still maintain $m \geq 32$, as so much is recommended by NIST [22]. Henceforth, it will be assumed that $m = 32$ and $b_s = 128$ bits, so that TESLA overhead is 160 bits per AE.

E. TESLA Security Issues

Although TESLA, and many modifications of it, have been discussed in the literature, no TESLA-like protocol has been standardized, and thus security concerns linger [23], [24]. However, this paper will assume the existence of a secure TESLA-like scheme, while acknowledging that significant cryptanalysis is necessary before a prudent civil GPS implementation. Even with secure TESLA, a TESLA-only NMA implementation has a glaring security issue: TESLA requires loose time synchronization. If a user has a time estimation error $|\delta_r| > \delta$, an attacker can field navigation data spoofing attacks by signing arbitrary data with old keys. The challenge of secure timing rests in clock initialization, after which the local clocks of GPS receivers can securely keep approximate time. When implemented within a client-server architecture, TESLA sessions can establish time and key chain parameters using digital signatures [20]. If altered for application within a one-way communication environment (e.g., by continually broadcasting signed initialization data), this initialization method would not resist replay attacks. Due to this issue, TESLA, acting on its own, is not a suitable data authentication technique for applications in which receiver time is derived from GPS signals, nor applications that wish to avoid dependence on secure network timing.

F. Comparison of ECDSA and TESLA

A thorough comparison of ECDSA and TESLA reveals differences in the computational cost for signal authentication via NMA: verifying a digital signature is more expensive than computing a MAC tag. However, this comparison is immaterial to algorithm choice as signature verification calculations are insignificant compared to typical GPS receiver signal processing [14].

Consider how selection among these two varieties of NMA (i.e, ECDSA- and TESLA-based) affects users with varying time estimation accuracy. ECDSA and TESLA both offer data authentication to users whose timing accuracy δ_r satisfies $|\delta_r| < \delta$. However, only ECDSA remains secure when $|\delta_r| > \delta$. Upon startup, receivers without an alternate time source will have large timing uncertainty and will not be able to use TESLA for data authentication. Note that while data authentication ensures navigation data are genuine, it does not prevent an attacker from replaying old data. Additionally, signal authentication requires μs-accurate timing, regardless of cryptographic method selection [14], [15].

G. Hybrid ECDSA-TESLA NMA

TESLA-based NMA has lower overhead than ECDSA-based NMA, despite the two varieties being equally secure for users with $|\delta_r| < \delta$. By contrast, only ECDSA-based NMA provides data authentication to users with poor time estimates $|\delta_r| > \delta$. Reference [14] selected a single cryptographic method for NMA. With the restriction to a single cryptographic method, it is necessary to select ECDSA over TESLA to uphold NMA's claim of cryptographic data authentication. However, this paper removes that restriction and considers schemes that employ both ECDSA and TESLA. For the same reason that ECDSA is chosen in the single-cryptographic- method case, all blended schemes must include ECDSA signatures, although TESLA components are optional. This paper proposes *hybrid NMA* to achieve both

- low-overhead, high-performance NMA for users with an alternate time source or already-acquired GPS signals, and
- cryptographic data authentication for all users.

In hybrid NMA, data for k consecutive TESLA MAC-key pairs is broadcast before the insertion of an ECDSA signature. The parameter k adjusts the mixture of TESLA and ECDSA. As k increases, the hybrid scheme relies more heavily on TESLA. Out of every $k+1$ AEs, only one—which corresponds to the ECDSA signature—provides cryptographic data authentication to users with $|\delta_r| > \delta$. While TESLA MACs authenticate data back to the previous AE (of any type), each ECDSA signature authenticates all data since the last ECDSA signature. Thus, all navigation data are signed by ECDSA. Figure 1 illustrates the interleaving of TESLA MAC-key pairs with ECDSA digital signatures for the $k = 2$ case. Note that $k = 0$ is a degenerate case in which there are only ECDSA AEs.

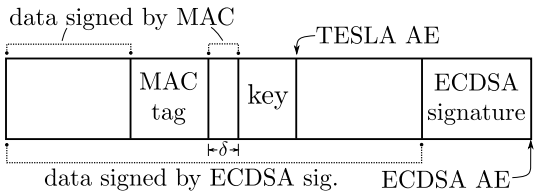

Fig. 1. Conceptual diagram of $k = 1$ hybrid NMA data stream.

The proposed $k > 0$ hybrid scheme ensures navigation data authenticity for all users while greatly reducing overhead compared to the ECDSA-only solution.

III. Transmission with GPS CNAV

This section explores specific formulations of the hybrid scheme for the GPS CNAV messaging structure. Let T_{ba} be the time between two consecutive AEs. If T_{ba} was to vary over time, a patient adversary would wait until T_{ba} took its worst-case value before launching an attack. For this reason, the most efficient pattern for AEs is perfectly periodic in time, so that T_{ba} is constant. Hybrid NMA is structured as $(k+1)T_{\text{ba}}$-second blocks, such as the $k = 1$ example in Fig. 1. AEs

occur periodically (i.e., T_{ba} is constant), even though the AE type may vary from one AE to the next.

A. CNAV Specifications

A CNAV message carries a 238-bit payload, excluding headers and integrity checks, and is broadcast on L2C and L5 signals over 12 seconds and 6 seconds, respectively, if forward error correction is enabled—an assumption that is made in this paper. For simplicity, this paper gives all durations under the assumption that NMA is implemented on the L2C signal; for an implementation on the L5 signal, all time durations (e.g., T_{ba}) will be reduced by factor two. The GPS Interface Specification specifies CNAV message types (MTs) and their maximum broadcast intervals [4], [25]. These maximum transmission intervals (for the L2C signal only) are shown in Table I considering both minimal and maximal broadcast conditions. The minimal broadcast is the transmission of only those MTs required under the Interface Control Working Group's planned changes to the GPS Interface Specification [26]. The maximal broadcast includes every MT enumerated in [4]. In both cases, MTs are assumed to be transmitted at the maximum transmission interval stated in [4].

TABLE I
TRANSMISSION INTERVALS FOR L2C CNAV MTS UNDER BOTH MINIMAL AND MAXIMAL BROADCAST CONDITIONS

MT	*Contents*	*Minimal*	*Maximal*	*Unallocated*
10	Ephemeris 1	48 sec.	48 sec.	3 bits
11	Ephemeris 2	48 sec.	48 sec.	7 bits
3*	Clock	48 sec.	48 sec.	up to 149 bits
30	Clock, ISC/IONO	288 sec.	288 sec.	12 bits
33	Clock, UTC	288 sec.	288 sec.	51 bits
35	Clock, GGTO	N/A	288 sec.	81 bits
32	Clock, EOP	N/A	30 min.	N/A
37	Clock, Midi Alm.	N/A	32 per 120 min.	N/A
31	Clock, Red. Alm.	N/A	20 min.	N/A
12	Reduced Alm.	N/A	4 per 20 min.	N/A
13	Diff. Corrections	N/A	30 min.	N/A
14	Diff. Corrections	N/A	30 min.	N/A

The entry 'MT-3*' in Table I reflects the requirement that every 48 seconds a message is broadcast that contains SV clock corrections, but the message may carry a secondary payload. Consider two new message types,

- MT-3X, which contains SV clock parameters and 149 bits of NMA data, and
- MT-YY contains 238 bits of NMA data.

B. Exploitation of Unallocated Bits

In addition to transmission of MT-3X and MT-YY, NMA data can be broadcast using N_e bits within the reserved space of existing MTs. The final column of Table I indicates the number of unallocated bits in each frequently-broadcast MT. Given the minimal broadcast requirements, N_e unallocated bits can be used for NMA during a $(k+1)T_{\text{ba}}$-second block provided $N_e \leq \left\lfloor 63\frac{(k+1)T_{\text{ba}}}{288} \right\rfloor + \left\lfloor 10\frac{(k+1)T_{\text{ba}}}{48} \right\rfloor$, where $\lfloor\cdot\rfloor$ is the floor function. For example, a total of $N_e = 123$ bits are available over 288 seconds of minimal CNAV broadcast by exploiting the unallocated bits in the following messages: one MT-30, one MT-33, four MT-10s, and four MT-11s.

C. Message Slots

The transmission requirements in Table I suggest four-message groups consisting of MT-10, MT-11, any MT that includes clock corrections (i.e., MT-3*), and any fourth message. These irreducable groups, illustrated in Fig. 2, constrain the placement of NMA messages. Since the fourth message in the group can be any MT, it will be referred to as an arbitrary message slot. Similarly, the choice of MT including clock corrections will be referred to as a clock message slot.

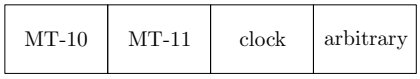

Fig. 2. Irreducible group of four contiguous messages required by CNAV specifications

D. Cost Metrics

To aid the selection of design parameters, several cost and performance metrics will be defined. One metric for NMA communication overhead is the fraction of CNAV bits used for NMA, or the *raw data fraction*. Although raw data fraction is informative, the reduction in CNAV flexibility caused by NMA is better measured by the number of message slots occupied by NMA data. For example, adding an NMA message containing a single allocated bit causes similar impact on CNAV flexibility as adding one that occupies all 238 payload bits. Therefore, let the *open data fraction* (ODF) be defined as the fraction of clock and arbitrary messages slots consumed by NMA, weighted by the unallocated payload size of clock and arbitrary messages. Over a $(k+1)T_{\text{ba}}$-second block, let N_{clk} and N_{arb} be the number of NMA-occupied clock and arbitrary message slots, respectively, and O_{clk} and O_{arb} be the number of open clock and arbitrary message slots after all guaranteed non-NMA messages have been schedules under the current broadcast condition. Then ODF is given as

$$\text{ODF} = \frac{149N_{\text{clk}} + 238N_{\text{arb}}}{149O_{\text{clk}} + 238O_{\text{arb}}} \text{ if } N_{\text{clk}} \leq O_{\text{clk}}, N_{\text{arb}} \leq O_{\text{arb}}.$$

For 288-second blocks and the minimal broadcast, $O_{\text{clk}} = 4$ and $O_{\text{arb}} = 6$, while during the maximal broadcast, $O_{\text{clk}} = 1.32$ and $O_{\text{arb}} = 4.72$. For given values of k, N_e, O_{clk}, and O_{arb}, one finds optimal N_{clk} and N_{arb} values by solving the optimization problem

$$\begin{aligned}
[N_{\text{clk}}, N_{\text{arb}}] &= \arg \min_{[N_{\text{clk}}, N_{\text{arb}}] \in \mathbb{S}} \text{ODF} \\
\mathbb{S} = \{[N_{\text{clk}}, N_{\text{arb}}] \in \mathbb{N}^2 \;\mid\; & N_{\text{clk}} + N_{\text{arb}} \geq k+1, \\
& 149N_{\text{clk}} + 238N_{\text{arb}} + N_e \geq 512 + 160k, \\
& N_{\text{clk}} \leq O_{\text{clk}}, \\
& N_{\text{arb}} \leq O_{\text{arb}}\},
\end{aligned}$$

where $\mathbb{N}$ is the set of non-negative integers.

E. Performance Metrics

In addition to minimizing cost, a hybrid NMA design should maximize a user's ability to thwart a GPS spoofing attack. There are two modifiable characteristics of hybrid NMA that strongly affect an attacker's capability: (1) T_{ba}, the time between consecutive AEs, and (2) the rate of unpredictable bits available to use for the detection of the security code estimation and replay (SCER) attacks described in [15]. The rate of unpredictable bits is equivalent to raw data fraction, and is thus unwise to maximize. Instead, any available bits, including those unallocated in existing MTs, should be populated with unpredictable bits to maximize SCER detection power. It is important to acknowledge that the T_{ba} definition treats all AEs equally, although only ECDSA signatures are applicable to users with poor time estimates $|\delta_r| > \delta$. The time between ECDSA-based AEs is $(k+1)T_{\text{ba}}$.

Define T_{fa} as the *time to first data authentication* for a user with a valid public key certificate (PKC). For users with approximate time $|\delta_r| < \delta$ at startup, T_{fa} is bounded as $\underline{T}_{\text{fa,n}} < T_{\text{fa}} < \bar{T}_{\text{fa,n}}$. Similarly, T_{fa} is bounded as $\underline{T}_{\text{fa,a}} < T_{\text{fa}} < \bar{T}_{\text{fa,a}}$ for users with insuffucient time accuracy $|\delta_r| > \delta$. The limits on T_{fa} are calculated by finding the smallest and largest window of navigation data necessary for an ECDSA AE, noting that $N_{\text{clk}} + N_{\text{arb}} = k+1$ ensures that only one NMA message is used for each TESLA AE, and assuming that the N_e bits are distributed throughout the block:

$$\underline{T}_{\text{fa,n}} = \begin{cases} T_{\text{ba}} & \text{if } N_e > 0 \\ 12 & \text{if } N_e = 0, N_{\text{clk}} + N_{\text{arb}} = k+1 \\ 24 & \text{if } N_e = 0, N_{\text{clk}} + N_{\text{arb}} > k+1 \end{cases}$$

$$\bar{T}_{\text{fa,n}} = \begin{cases} 3T_{\text{ba}} & \text{if } N_e > 0 \\ 2T_{\text{ba}} + 12 & \text{if } N_e = 0, N_{\text{clk}} + N_{\text{arb}} = k+1 \\ 3T_{\text{ba}} & \text{if } N_e = 0, N_{\text{clk}} + N_{\text{arb}} > k+1 \end{cases}$$

$$\underline{T}_{\text{fa,a}} = \begin{cases} (k+1)T_{\text{ba}} & \text{if } N_e > 0 \\ kT_{\text{ba}} + 12 & \text{if } N_e = 0, N_{\text{clk}} + N_{\text{arb}} = k+1 \\ kT_{\text{ba}} + 24 & \text{if } N_e = 0, N_{\text{clk}} + N_{\text{arb}} > k+1 \end{cases}$$

$$\bar{T}_{\text{fa,a}} = \begin{cases} 2(k+1)T_{\text{ba}} & \text{if } N_e > 0 \\ (2k+1)T_{\text{ba}} + 12 & \text{if } N_e = 0, N_{\text{clk}} + N_{\text{arb}} = k+1 \\ 2(k+1)T_{\text{ba}} & \text{if } N_e = 0, N_{\text{clk}} + N_{\text{arb}} > k+1. \end{cases}$$

For example, a scheme with $N_e = 0$ and one message per TESLA AE offers a user with alternate time source a low $\underline{T}_{\text{fa,n}}$ of 12 seconds. In this scenario, a networked receiver that begins tracking a signal just before a TESLA NMA message starts can authenticate the signal after just 12 seconds. Further analysis will focus on T_{ba} performance, while acknowledging that both large k and $N_e > 0$ hinder T_{fa} performance.

F. Parameter Selection

The design process consists of modifying three parameters of hybrid NMA (T_{ba}, k, N_e) and observing the resulting cost metric (ODF). A subset of this search is in Fig. 3, which sweeps over k for various T_{ba} and N_e values, and in Fig. 4, which sweeps over T_{ba} for $N_e = 0$ and $k \in \{0, 2, 5\}$.

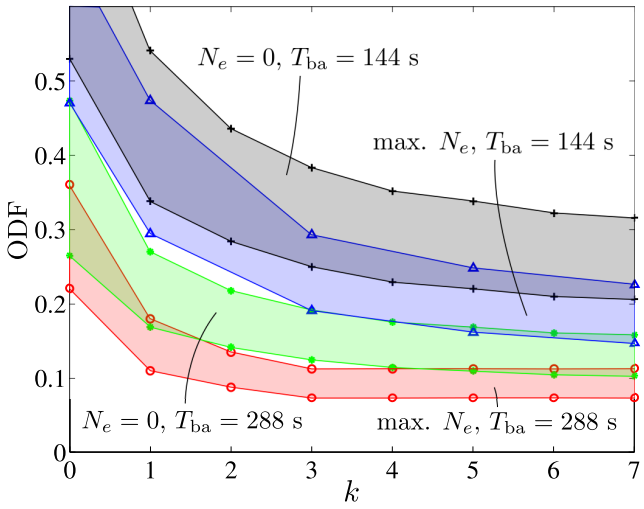

Fig. 3. Open data fraction for various hybrid NMA schemes. In each pair of traces, the lower trace assumes the minimal CNAV broadcast while the upper trace assumes the maximal broadcast.

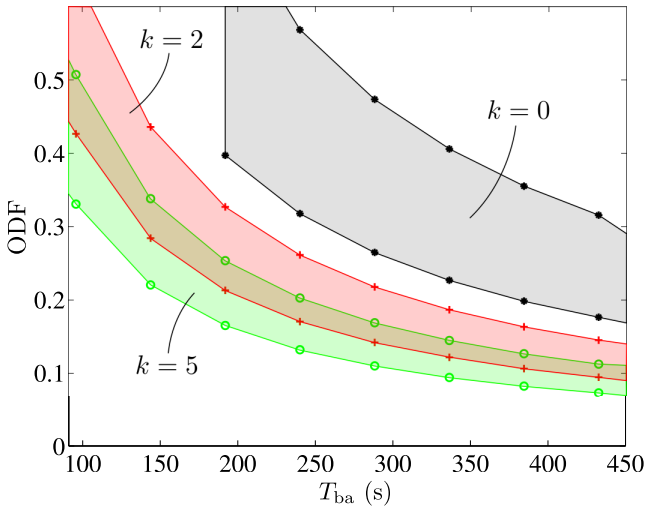

Fig. 4. Open data fraction for hybrid NMA with various T_{ba} values. In each pair of traces, the lower trace assumes the minimal CNAV broadcast while the upper trace assumes the maximal broadcast.

IV. Example Payload Bit Specifications

A few example payload bit specifications for new NMA MTs are offered as concrete proposals drawn from the general discussion in the preceding section. When the number of allocated payload bits, $149N_{\text{clk}} + 238N_{\text{arb}} + N_e$, exceeds the required bits for hybrid NMA, $512 + 160k$, there are unallocated payload bits within the new NMA MTs. Additionally, if N_e is less than its maximum value, there are still unallocated bits in existing MTs which are not exploited for NMA. These unallocated bits can benefit NMA by being using to broadcast public key certificates (PKCs) or cryptographic salt (i.e., pseudo-random bits).

A. Public Key Certificates

A 424-bit format for hybrid NMA PKCs is proposed in Table II. The ECDSA curve identifier allows the GPS Control Segment to select amongst a small suite of elliptic curves, in case vulnerabilities are found with a specific curve. The start and end of validity fields indicate the time window over which the key is active. The key is activated (deactivated) at the start of the GPS week with week number four times the value of the start (end) of validity field. The TESLA algorithm identifier specifies a suite of functions to include the MAC tag generation method and the key derivation method. If $k = 0$, then only the first 276 bits of the PKC are necessary.

TABLE II
PROPOSED PUBLIC KEY CERTIFICATE FORMAT

bits	category	contents
1-2	ECDSA	curve identifier
3-11	ECDSA	start of validity
12-20	ECDSA	end of validity
21-276	ECDSA	public key
277-278	TESLA	algorithm identifier
279-287	TESLA	start of validity
288-296	TESLA	end of validity
297-424	TESLA	public key

For users without a secure side communication channel, an initial PKC must be inserted manually or during device manufacture, while subsequent PKCs can be delivered over-the-air in GPS navigation data. It is recommended that key exposure periods are limited to one to three years [17]. While it may seem reasonable to distribute PKCs shortly (e.g., a week) prior to the start of their validity, such a scheme would require that a non-networked receiver be active for the time preceding the key changeover. To facilitate practical over-the-air re-keying, PKCs should be distributed well before the start of their validity period. An example plan could be to disclose PKCs six months prior to their six month validity period. With this plan, there would be six months to distribute each PKC to all NMA users and each key exposure period would be one year. The rate of key distribution can be characterized by the minimum duration of navigation data reception required to ensure a user receives a complete PKC (e.g., users who possess two-hours-worth of navigation data in each six month window are guaranteed to receive all PKCs). PKCs can be included in the CNAV broadcast by either adding dedicated MTs or exploiting unallocated bits in existing or new NMA MTs.

B. NMA Message Design

As noted previously, the power of SCER attack detection tests depends on the rate of unpredictable symbols, which is equal to the rate of unpredictable bits both with and without forward error correction enabled. To aid these detection tests, unallocated bits in existing or new NMA MTs that are not used for PKC broadcast should be populated with salt. The following example specifications for new NMA MTs attempt to include PKC broadcast but do not reflect the recommendation to populate unallocated bits in existing MTs with salt.

The analysis of hybrid NMA in the preceding sections permits many possible parameter values. For brevity, a small sampling of these parameter values are selected as representative of important tradeoffs. In each example, definitions for the new CNAV MTs are proposed. While the example are practical solutions, full details on system constraints will likely suggest adoption of alternate parameter values.

1) Schemes with $k = 0$: If TESLA is not trusted for use in civil GPS NMA due to lack of standardization, it is necessary to set $k = 0$. Table III and Table IV show scheme A and scheme B, two example designs with $N_e = 0$ and the maximum N_e value, respectively. The ECDSA signature S is broken up into different messages and then re-assembled as the concatenation of the components S_i.

TABLE III
PAYLOAD BIT SPECIFICATION FOR SCHEME A: $k = 0$, $N_e = 0$

MT	bits	contents
3X	90-228	S_1 or S_2
	229	PKC start flag
	230-238	PKC$_l$
YY	1-234	S_3
	235-238	salt

TABLE IV
PAYLOAD BIT SPECIFICATION FOR SCHEME B: $k = 0$, $N_e = 123$, $T_{\text{BA}} = 288$

MT	bits	contents
3X	90-233	S_{15}, S_{16}, or S_{17}
	234-238	salt
10	236-238	S_{2i-1}, $i \in 1, \ldots, 6$
11	232-238	S_{2i}, $i \in 1, \ldots, 6$
33	188	PKC start flag
	189-230	PKC$_l$
	231-238	S_{13}
30	227-238	S_{14}

When $T_{\text{ba}} = 288$ seconds, scheme A has ODF of 25%-47%, depending on the current broadcast condition. Scheme B has a lower ODF of 22%-36%. A full PKC transmission takes, on average, 2.5 hours with scheme A and 34 minutes with scheme B.

2) Scheme with $k = 2$: To significantly reduce overhead, consider $k = 2$ so that only one-third of AEs use ECDSA signatures. Table V shows such a scheme with $N_e = 0$. To authenticate via TESLA, the condition $|\delta t_r| < 880$ milliseconds must hold so that the user can ensure MAC tags are received before the key used in their generation is disclosed. This 880-millisecond limit holds for scheme C and all subsequent schemes. The ODF of scheme C is 28%-44% when $T_{\text{ba}} = 144$ seconds, comparable to that of scheme A when $T_{\text{ba}} = 288$ seconds. As shown in Fig. 4, choosing a longer T_{ba} can lead to low ODF values, including the ODF range 14%-22% for $T_{\text{ba}} = 288$ seconds.

3) Schemes with $k = 5$: Overhead can be further reduced by increasing k to five and using scheme D from Table VI. Assuming $T_{\text{ba}} = 144$ seconds, scheme D has an ODF range of 22%-34% (compared to 28%-44% with scheme C). For

TABLE V
PAYLOAD BIT SPECIFICATION FOR SCHEME C: $k = 2$, $N_e = 0$

MT	bits	contents
XX	1-32	MAC tag
	33-92	S_1 or S_2
	93	PKC start flag
	94-110	PKC$_l$
	111-238	TESLA key
3Y	90-238	S_3
ZZ	1-233	S_4
	234-238	salt

large k and T_{ba}, the limits on T_{fa} become large. For example, selection of scheme D with $T_{\text{ba}} = 288$ seconds leads to $\bar{T}_{\text{fa,a}} = 53$ minutes. If T_{fa} for non-networked users is important, the product $(k+1)T_{\text{ba}}$ should be reduced.

TABLE VI
PAYLOAD BIT SPECIFICATION FOR SCHEME D: $k = 5$, $N_e = 0$

MT	bits	contents
XX	1-32	MAC tag
	33-37	salt
	38-110	S_i, $i \in 1, \ldots, 5$
	111-238	TESLA key
3Y	90-236	S_6
	237-238	salt

Note that since there are few unallocated bits within the NMA MTs of scheme D, over-the-air PKC broadcast is not explicitly provided—such data must be placed in the unallocated portions of existing MTs or in a MT dedicated to PKC broadcast. Consider scheme E (with $N_{\text{clk}} = 0$, $N_{\text{arb}} = 6$) as an alternative to scheme D (with $N_{\text{clk}} = 1$, $N_{\text{arb}} = 5$). Shown in Table VII, scheme E is not ODF-optimal, but allows for over-the-air PKC broadcast every 50 minutes, assuming $T_{\text{ba}} = 144$ seconds.

TABLE VII
PAYLOAD BIT SPECIFICATION FOR SCHEME E: SUB-OPTIMAL VERSION OF $k = 5$, $N_e = 0$

MT	bits	contents
XX	1-32	MAC tag
	33-88	S_i, $i \in 1, \ldots, 5$
	89	PKC start flag
	90-110	PKC$_l$
	111-238	TESLA key
YY	1-232	S_6
	233-238	salt

V. CONCLUSIONS

A hybrid ECDSA-TESLA navigation message authentication (NMA) scheme was proposed for the modernized GPS civil navigation (CNAV) message. The scheme was analyzed to determine cost, as measured by impact on CNAV message stream availability, and performance, as measured by time between authentications and time to first authentication. The scheme improved upon previous NMA proposals by drastically reducing overhead while preserving cryptographic authentication of navigation data for all users. New CNAV message types were defined for a few scenarios which illustrate cost-performance tradeoffs.

REFERENCES

[1] 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.

[2] D. P. Shepard, T. E. Humphreys, and A. A. Fansler, "Evaluation of the vulnerability of phasor measurement units to GPS spoofing attacks," *International Journal of Critical Infrastructure Protection*, vol. 5, no. 3-4, pp. 146–153, 2012.

[3] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via GPS spoofing," *Journal of Field Robotics*, 2014. to be published.

[4] GPS Directorate, "Systems engineering and integration Interface Specification IS-GPS-200G," 2012. http://www.gps.gov/technical/icwg/.

[5] K. D. Wesson, B. L. Evans, and T. E. Humphreys, "A combined symmetric difference and power monitoring GNSS anti-spoong technique," in *IEEE Global Conference on Signal and Information Processing*, 2013.

[6] M. Psiaki, B. O'Hanlon, J. Bhatti, D. Shepard, and T. Humphreys, "GPS spoofing detection via dual-receiver correlation of military signals," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2250–2267, 2013.

[7] B. O'Hanlon, M. Psiaki, J. Bhatti, and T. Humphreys, "Real-time spoofing detection using correlation between two civil GPS receiver," in *Proceedings of the ION GNSS Meeting*, (Nashville, Tennessee), Institute of Navigation, 2012.

[8] D. S. D. Lorenzo, J. Gautier, J. Rife, P. Enge, and D. Akos, "Adaptive array processing for GPS interference rejection," in *Proceedings of the ION GNSS Meeting*, (Long Beach, CA), Institute of Navigation, Sept. 2005.

[9] P. Y. Montgomery, T. E. Humphreys, and B. M. Ledvina, "A multi-antenna defense: Receiver-autonomous GPS spoofing detection," *Inside GNSS*, vol. 4, pp. 40–46, April 2009.

[10] A. Broumandan, A. Jafarnia-Jahromi, V. Dehgahanian, J. Nielsen, and G. Lachapelle, "GNSS spoofing detection in handheld receivers based on signal spatial correlation," in *Proceedings of the IEEE/ION PLANS Meeting*, (Myrtle Beach, SC), Institute of Navigation, April 2012.

[11] L. Scott, "Anti-spoofing and authenticated signal architectures for civil navigation systems," in *Proceedings of the ION GNSS Meeting*, pp. 1542–1552, 2003.

[12] G. Becker, S. Lo, D. De Lorenzo, D. Qiu, C. Paar, and P. Enge, "Efficient authentication mechanisms for navigation systems—a radio-navigation case study," in *Proceedings of the ION GNSS Meeting*, (Savannah, Georgia), Institute of Navigation, 2009.

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

[14] K. D. Wesson, M. P. Rothlisberger, and T. E. Humphreys, "Practical cryptographic civil GPS signal authentication," *Navigation, Journal of the Institute of Navigation*, vol. 59, no. 3, pp. 177–193, 2012.

[15] T. E. Humphreys, "Detection strategy for cryptographic GNSS anti-spoofing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1073–1090, 2013.

[16] M. Luk, A. Perrig, and B. Whillock, "Seven cardinal properties of sensor network broadcast authentication," in *Proceedings of the ACM workshop on security of ad hoc and sensor networks*, (Alexandria, VA), pp. 147–156, Oct. 2006.

[17] NIST, "Recommendation for key management—Part I: General (revised)," SP 800-57, National Institute of Standards and Technology, July 2012.

[18] NIST, "Digital signature standard," FIPS PUB 186-4, National Institute of Standards and Technology, July 2013.

[19] ECC Brainpool, "ECC Brainpool standard curves and curve generation," tech. rep., Elliptic Curve Cryptography Brainpool, Oct. 2005.

[20] A. Perrig, R. Canetti, D. Song, and J. Tygar, "Efficient and secure source authentication for multicast," in *Proceedings of the Network and Distributed System Security Symposium*, Internet Society, Feb. 2001.

[21] NIST, "The keyed-hash message authentication code," FIPS PUB 198-1, National Institute of Standards and Technology, July 2008.

[22] Q. Dang, "Recommendation for applications using approved hash algorithms (revised)," SP 800-107, National Institute of Standards and Technology, Aug. 2007.
[23] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, and D. Culler, "SPINS: Security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.
[24] G. Jakimoski, "Some notes on the security of the timed efficient stream loss-tolerant authentication scheme," in *Selected Areas in Cryptography*, pp. 342–357, Springer Berlin Heidelberg, 2007.
[25] GPS Directorate, "Systems engineering and integration Interface Specification IS-GPS-705C," 2012. http://www.gps.gov/technical/icwg/.
[26] GPS Directorate, "Clarification of CNAV broadcast intervals in IS-GPS-200G," 2013. http://www.gps.gov/technical/icwg/meetings/2013/.