

GNSS signal authentication

– some applications and implementation options

Markus Kuhn



UNIVERSITY OF
CAMBRIDGE

Computer Laboratory

<http://www.cl.cam.ac.uk/~mgk25/>

Spoofing navigation signals has a long history

“Wreckers” or “mooncussers” faking light-tower signals to lure cargo ships into dangerous waters and steal cargo from the wreck



“Mooncussers on rock with lantern”

Brenda Z. Guiberson: *Lighthouses: Watchers at Sea*, 1995

GNSS signal authentication

In a hostile environment, a GNSS receiver ought to be able to distinguish whether its RF input signal is

- ▶ **genuine** – received straight from the expected satellite
- ▶ **spoofed** – synthesized by a signal generator, alternated by a signal processing system, not propagated along a straight line

GNSS signal authentication

In a hostile environment, a GNSS receiver ought to be able to distinguish whether its RF input signal is

- ▶ **genuine** – received straight from the expected satellite
- ▶ **spoofed** – synthesized by a signal generator, alternated by a signal processing system, not propagated along a straight line

In military applications: main worry are spoofed RF signals emitted at a distance, to redirect or confuse bombs, soldiers, or vehicles elsewhere.

→ “Navigation warfare” (NAVWAR)

- ▶ Local antenna trusted
- ▶ Spoofer interferes from a distance

GNSS signal authentication

In a hostile environment, a GNSS receiver ought to be able to distinguish whether its RF input signal is

- ▶ **genuine** – received straight from the expected satellite
- ▶ **spoofed** – synthesized by a signal generator, alternated by a signal processing system, not propagated along a straight line

In military applications: main worry are spoofed RF signals emitted at a distance, to redirect or confuse bombs, soldiers, or vehicles elsewhere.

→ “Navigation warfare” (NAVWAR)

- ▶ Local antenna trusted
- ▶ Spoofer interferes from a distance

In some civilian applications: → Remote attestation

- ▶ person in possession of GNSS receiver is not trusted
- ▶ Antenna manipulated, replaced, covered
- ▶ Spoofer controls RF-input port of GNSS receiver

Existing real-word sensor attacks

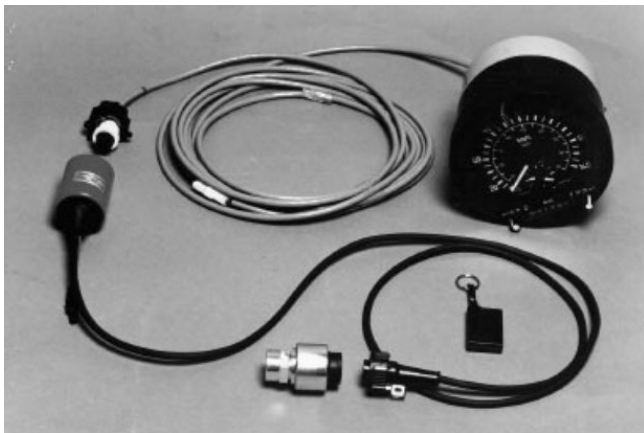


Photo: Hampshire Constabulary / Ross Anderson

PVT-sensor spoofing devices have already been found “in the wild” by British police: in commercial good vehicles between tachograph and gearbox sensor. Drivers use them to manipulate their velocity and working-hours record.

Secure positioning – cryptography at light speed

Conventional cryptographic authentication protocols

- ▶ establish the identity of communication partners
- ▶ protect the integrity of data

Secure positioning – cryptography at light speed

Conventional cryptographic authentication protocols

- ▶ establish the identity of communication partners
- ▶ protect the integrity of data

but do not authenticate

- ▶ the location of communication partners
- ▶ the nanosecond-resolution transmission time of data

Secure positioning – cryptography at light speed

Conventional cryptographic authentication protocols

- ▶ establish the identity of communication partners
- ▶ protect the integrity of data

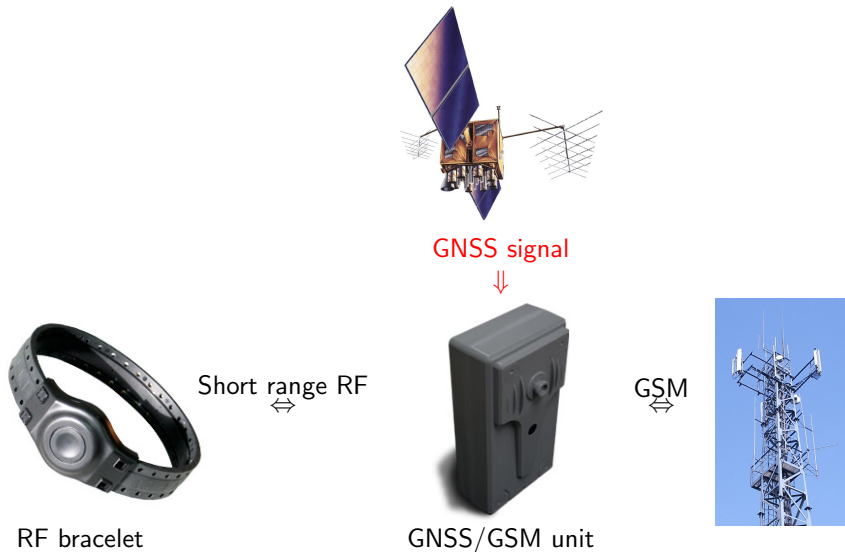
but do not authenticate

- ▶ the location of communication partners
- ▶ the nanosecond-resolution transmission time of data

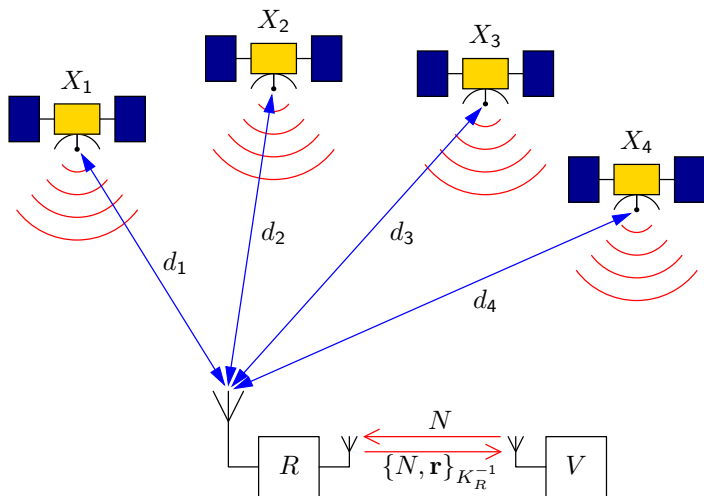
Protection technologies:

- ▶ Asymmetric security for satellite navigation signals
- ▶ Two-way distance-bounding protocols
- ▶ Tamper-resistant hardware

Example application: offender tagging



Remote attestation of position



Remotely-queried navigation-signal receiver R is a trusted component, in the hands of someone (thief, electronic prisoner, road-tax avoider) who wants it to report a *pretended position* $\mathbf{r}'$ instead of its *actual position* $\mathbf{r}$.

Global positioning systems in future cars

GNSS receivers are becoming a standard feature in new cars.

Primary applications:

- ▶ route finding, service location
- ▶ automatic emergency calls

Service to the driver, no tamper-resistance requirement

Secondary applications:

- ▶ fleet management
- ▶ usage-based car insurance
- ▶ usage-based road tax
- ▶ congestion charging
- ▶ speed-limit enforcement
- ▶ theft protection
- ▶ forensic reconstruction of accidents, alibi verification, . . .

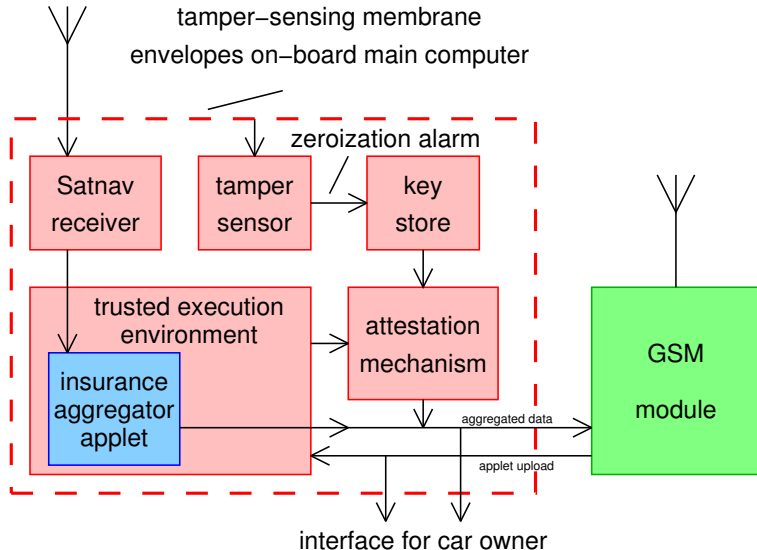
Potential legislative/contractual requirement, adversarial user, tamper-resistance requirement

Use-based car insurance

First deployment of “tamper resistant” GPS in private cars

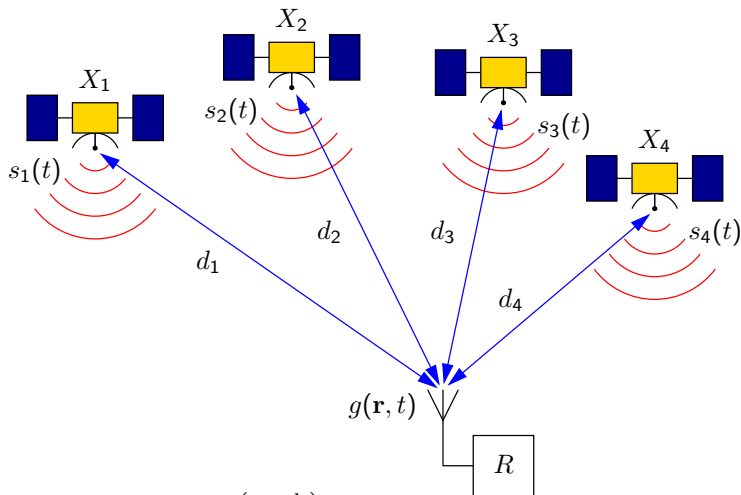
- ▶ pay-as-you-drive or pay-how-you-drive policies:
 - US (Progressive Insurance Snapshot, MetroMile, etc.)
 - Italy (Octo Telematics)
 - Spain (Telefónica Insurance Telematics)
 - Germany (Sparkasse S-Drive-Service)
 - Discontinued: UK (Norwich Union), Ireland (AXA “Traksure”)
- ▶ mileage during peak and off-peak hours → transfer via GSM
- ▶ currently an add-on GPS box provided by insurance company
- ▶ later integrated with normal onboard computer network
Progressive’s “TripSense” OBD-II module is a first step in that direction
- ▶ eventually merely a 3rd-party software applet?
 - standardized car operating-system API
 - compartmentalization and trusted computing features
- ▶ privacy concerns vs substantial insurance discounts

Remote attestation of aggregated position



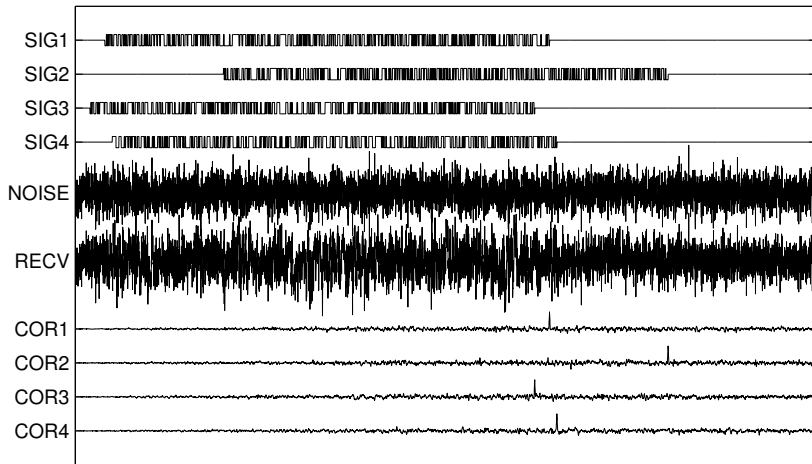
Privacy-friendly version: car owner can inspect data aggregator applet (simple fee spread sheet).

Pseudorange positioning systems



$$g(\mathbf{r}, t) = \sum_i A_i \cdot s_i \left(t - \frac{d_i}{c} \right) + n(\mathbf{r}, t)$$

Correlation receiver



$$C_i(\mathbf{r}, t) = \int g(\mathbf{r}, \tau) \cdot s(\tau - t) d\tau$$

Existing technology:

- ▶ GPS/Galileo open access channel:
everyone can fake the satnav signal
- ▶ GPS military channel:
military receiver knows shared secret $\Rightarrow$ can fake signal
- ▶ Galileo subscription channel: need to break SIM to fake signal

Wanted: Asymmetric security

- ▶ Those who can verify the integrity of the signal cannot at the same time fake it.
- ▶ Public-key signatures provide this for data.
- ▶ But in navigation, **not only the data, but also its nanosecond relative arrival time must be protected** against manipulation (selective-delay attacks).

Message authentication code (MAC)

- ▶ $K \leftarrow \text{Gen}$ private-key generation
- ▶ $C \leftarrow \text{Mac}_K(M)$ MAC generation
- ▶ $\text{Vrfy}_K(M', C) = 1$ MAC verification
 $\Leftrightarrow M \stackrel{?}{=} M'$

Typical MAC lengths: 24...96 bit

Digital signature

- ▶ $PK, SK \leftarrow \text{Gen}$ public/secret key-pair generation
- ▶ $S \leftarrow \text{Sign}_{SK}(M)$ signature generation using secret key
- ▶ $\text{Vrfy}_{PK}(M', S) = 1$ signature verification using public key
 $\Leftrightarrow M \stackrel{?}{=} M'$

Typical signature length: 320 bits (ECDSA), can offer message recovery

One-way “hash” functions

A function h is *one way* if it is not computationally feasible to find for a given y any x such that $h(x) = y$.

Examples: SHA-1, SHA-2, SHA-3

Lamport hash chain

- ▶ Chose length n and secret random start value R_n (e.g. 80...128 bit)
- ▶ Generate $R_0, R_1, \dots, R_{n-1}$ with $R_{i-1} = h(R_i)$ or equivalently

$$R_i = \underbrace{h(h(h(\dots h(R_n)\dots)))}_{(n-i) \text{ times}} = h^{n-i}(R_n)$$

- ▶ Publish/reveal R_0 over authenticated channel

When R_1 is revealed, everyone who already knows R_0 can verify that R_1 is genuine via $R_0 = h(R_1)$. Similarly when R_2 is revealed, etc.

Timed Efficient Stream Loss-tolerant Authentication

TESLA uses a hash chain to authenticate broadcast data, without any need for a digital signature for each message.

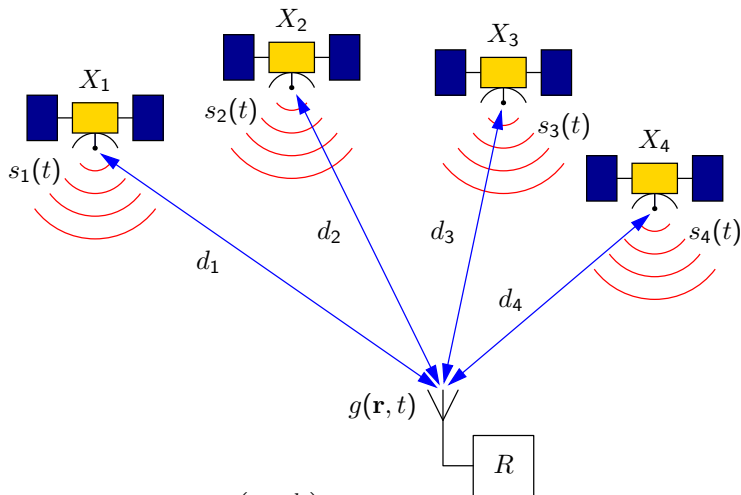
Timed broadcast of data sequence $M_1, M_2, \dots, M_n$:

- ▶ $t_0 : \text{Sign}(R_0), R_0$ where $R_0 = h(R_1)$
- ▶ $t_1 : (\text{Mac}_{R_2}(M_1), M_1, R_1)$ where $R_1 = h(R_2)$
- ▶ $t_2 : (\text{Mac}_{R_3}(M_1), M_2, R_2)$ where $R_2 = h(R_3)$
- ▶ $t_3 : (\text{Mac}_{R_4}(M_1), M_3, R_3)$ where $R_3 = h(R_4)$
- ▶ $t_4 : (\text{Mac}_{R_5}(M_1), M_4, R_4)$ where $R_4 = h(R_5)$
- ▶ ...

Each R_i is revealed at a pre-agreed time t_i . The MAC for M_i can only be verified after t_{i+1} when key R_{i+1} is revealed.

By the time the MAC key is revealed, everyone has already received the MAC, therefore the key can no longer be used to spoof the message.

Pseudorange positioning systems



$$g(\mathbf{r}, t) = \sum_i A_i \cdot s_i \left(t - \frac{d_i}{c} \right) + n(\mathbf{r}, t)$$

Synthesis of predictable signal

Attacker connects receiver to a signal generator that emulates – knowing the predictable waveforms $s_i(t)$ – the signal $g(\mathbf{r}', t)$, as it would be received at the pretended position $\mathbf{r}'$.

Countermeasure:

- ▶ Add to $s_i(t)$ an unpredictable but verifiable element, e.g. include a message authentication code or digital signature of the current time and navigation message.

Selective-delay attack

Attacker uses signal $g(\mathbf{r}, t)$ at the actual position $\mathbf{r}$ and converts it into a prediction of the signal $g(\mathbf{r}', t - \Delta t)$ that would have been received at the pretended position $\mathbf{r}'$ a short time Δt earlier, and feeds that into the receiver.

Countermeasure:

- ▶ Receiver needs a trusted local clock precise enough to detect delay Δt .

Relaying attack

Disconnect R from its antenna and connect it via a communication link to a remote antenna at pretended location $\mathbf{r}'$.

Less likely, since

- ▶ challenging logistics for attacker
- ▶ remote antenna easy to locate
- ▶ wideband signal may be difficult to relay with low latency

Making the navigation signal unpredictable

$$\text{Navigation signal} = \text{Spreading code} \times \text{navigation message}$$

GPS C/A signal:

- ▶ Spreading code = 1.023 Mbit/s (chips) repeating every 1ms
- ▶ Navigation message = 50 bit/s

Equivalent distances:

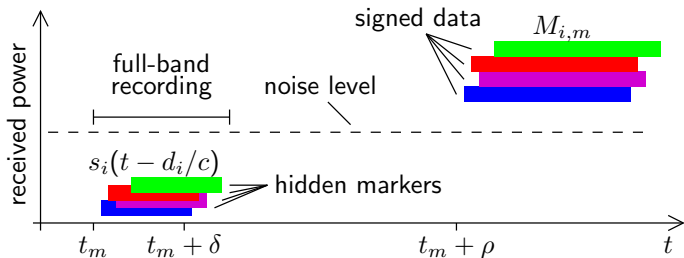
- ▶ 1 chip = 1 μ s = 300 m
- ▶ 1 PRN cycle = 1 ms = 300 km
- ▶ 1 bit = 20 ms = 6000 km

Galileo OS similar: ≈ 2 Mbit/s, 200 bit/s

Asymmetric satnav security through hidden markers:

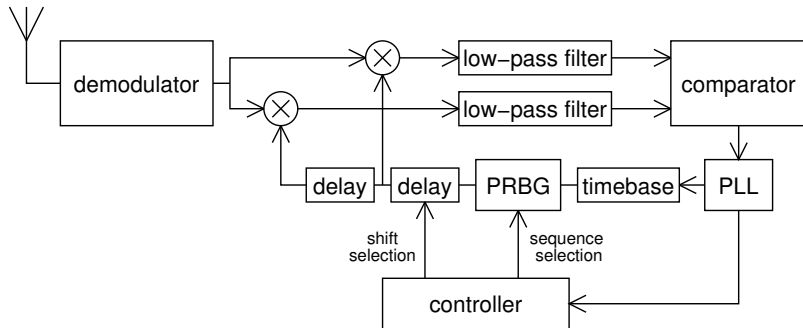
This solution is a steganographic process:

- ▶ transmitters broadcast unpredictable spread-spectrum carrier below noise threshold
- ▶ receivers record full bandwidth
- ▶ transmitters release random-noise seed after a delay ρ
- ▶ receivers use FFT-based convolution to detect hidden markers



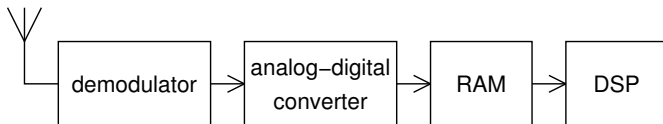
GPS receiver

Simplified operation of a traditional GPS receiver:



A local timebase drives a local random-bit generator, which a PLL controlled by a real-time early/late correlator keeps phase-locked with the transmitter's timebase. The controller switches between the sequences of different satellites and adjusts/records their relative delay.

Delayed correlation receiver:



Basic idea

- ▶ Every few seconds, all transmitters broadcast a *hidden marker*.
- ▶ A hidden marker carries no data.
- ▶ It is an unpublished spreading sequence broadcast at least 30 dB below the minimal noise seen by any receiver.
- ▶ Receivers digitize and buffer in RAM the full bandwidth of the hidden markers while they are broadcast. This preserves their relative arrival times, but it cannot be accessed yet.
- ▶ After a delay ρ , the transmitters broadcast the seed value used to generate the hidden marker, which was secret until then.
- ▶ Receivers (and attackers!) can only now identify and separate the markers in the recorded antenna signal.

A signal-synthesis or selective-delay attack can now be performed only with a delay $\Delta t > \rho$.

Choose ρ large enough (e.g., 10 s), such that even receivers with a cheap clock can discover the delay in the received timestamps.

Attacks with directional antennas

Problem: Attacker could use four directional antennas that track the satellites to isolate their signals (for a selective-delay attack).

- ▶ If antenna gain is high enough to lift signal out of noise, it can be made noise-free with a threshold operator.
- ▶ Otherwise, attacker can still delay and mix the four antenna signals, without removing their noise.

Potential countermeasure: No directional antenna is perfect.

- ▶ Attenuated residual signals from all transmitters will be present in each antenna signal.
- ▶ If these show up as secondary peaks in the cross correlation
⇒ selective-delay attack is in progress.
- ▶ Receiver rejects correlation results with too high secondary peaks.
- ▶ Maximum amplitude of secondary peak is a security parameter that determines attack cost.

Unpredictable data bits: a medium-term compromise?

A GNSS signal today is the product of a fast spreading code (1–10 Mbit/s) and a slow navigation data stream (50–500 bit/s).

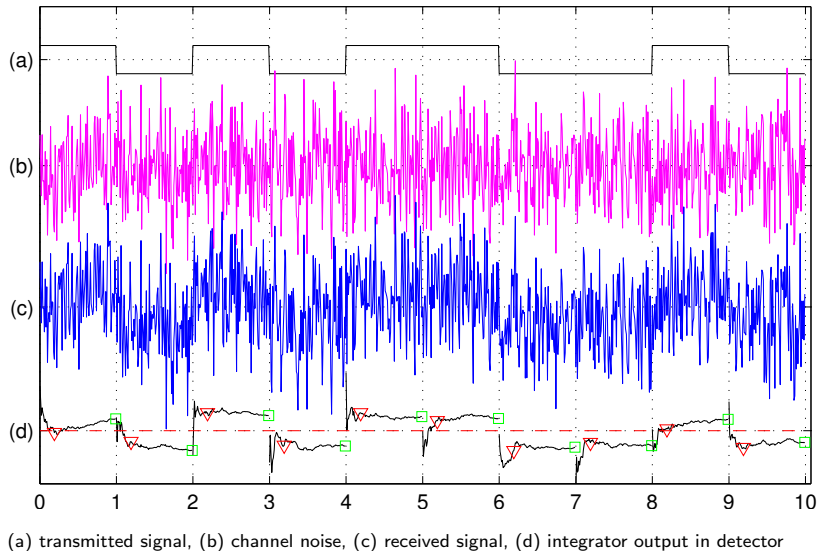
Switching to an initially secret spreading code is a major change to a GNSS system, not easily achieved in the near future.

Alternative approach: adding unpredictable bits to the navigation message also creates unpredictable clock edges.

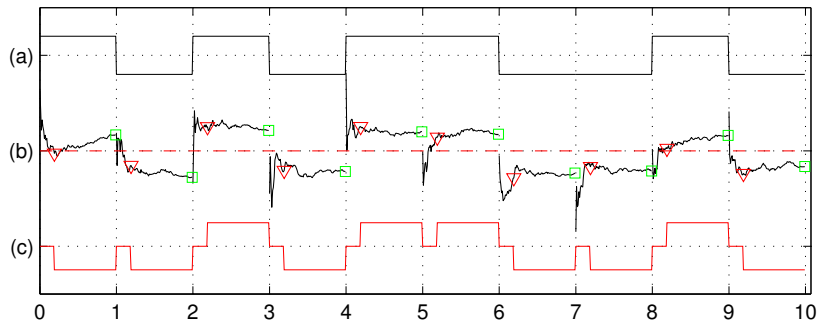
Problems:

- ▶ unpredictable bits appear at a significantly lower rate (10^6)
- ▶ an attacker can perform early detection
 - a normal receiver detects data bits using a matched filter that integrates the signal over the full duration of the data bit → lowest bit error rate
 - a spoofer can operate a receiver that detects data bits after integrating only an initial fraction of the duration of the data bit → earlier knowledge at the cost of higher bit error rate
 - a spoofer can then adjust the data value in the spoofed signal for the remaining duration of the bit period

Special detector provides early bit estimate



Special detector provides early bit estimate



(a) transmitted signal, (b) integrator output in detector, (c) generated spoofed signal

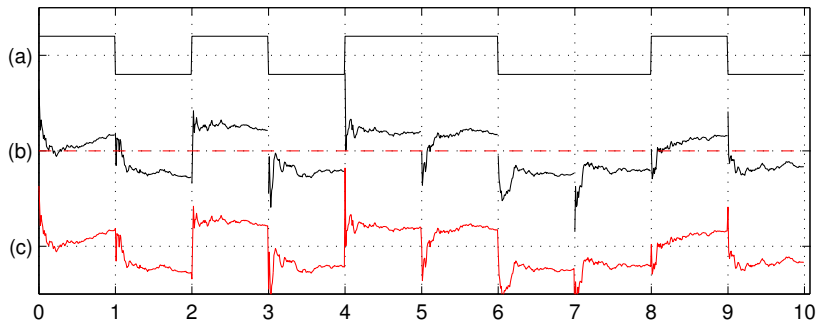
Early decision:

$$b(t) = (t - t_0)^{-1} \cdot \int_{t_0}^t a(t) dt$$

- ▶ At start of bit (t_0), output neutral value 0
- ▶ After fraction x of the bit duration T has passed ($t > t_0 + xT$), output

$$\text{sgn}(b(t_0 + xT)) \cdot (1 - x)^{-1}$$

Special detector provides early bit estimate



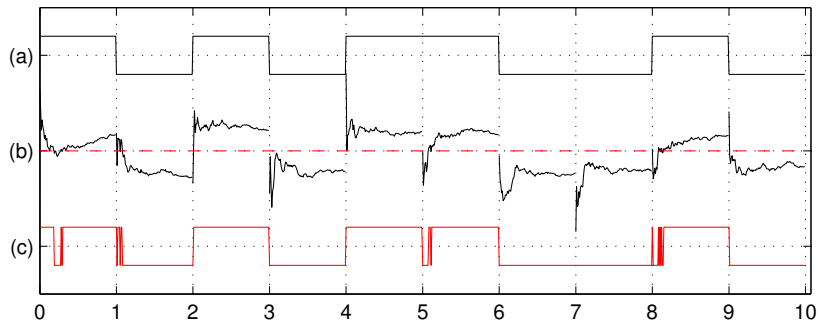
(a) transmitted signal, (b) integrator output in detector, (c) generated spoofed signal

Maximum likelihood estimate (ML): $b(t) = (t - t_0)^{-1} \cdot \int_{t_0}^t a(t) dt$

- Continuously output current integrator value:

$$b(t)$$

Special detector provides early bit estimate



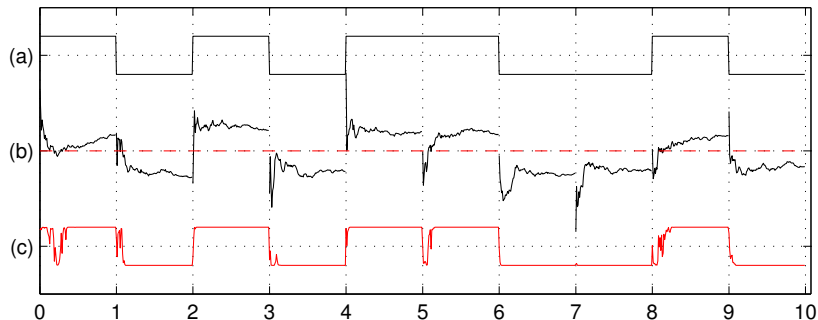
(a) transmitted signal, (b) integrator output in detector, (c) generated spoofed signal

Max. a posteriori prob. estimate (MAP): $b(t) = (t - t_0)^{-1} \cdot \int_{t_0}^t a(t) dt$

- Continuously apply threshold to current integrator value:

$$\text{sgn}(b(t))$$

Special detector provides early bit estimate



(a) transmitted signal, (b) integrator output in detector, (c) generated spoofed signal

Min. mean-square error estimate (MMSE): $b(t) = (t - t_0)^{-1} \cdot \int_{t_0}^t a(t) dt$

► Continuously output:

$$\tanh(b(t)/\sigma^2(t))$$

[Humphreys 2013]

Unpredictable data bits

An unpredictable data bit is only unknown to the spoofer during the first fraction of the bit period (e.g. tens of microseconds, depending on C/N) and revealed clearly immediately afterwards

- ▶ GNSS receiver needs highly accurate independent clock for authentication based on unpredictable bits
 - required accuracy not available from non-GNSS sources
 - data-based authentication only practical if receiver was recently tracking a genuine signal
 - in many remote-attestation applications, the spoofer may have full control over the antenna signal for week, right from cold boot
- ▶ adding random data bits complicates spoofer design somewhat (early detector, COTS receivers output data only at end of subframe), but ultimately no substitute for the steganographic solution
- ▶ random data bits best added in form of TESLA authentication of navigation message (prerequisite for steganographic solution)
- ▶ investigate adding separate steganographic channel, with deliberately low carrier power, aimed at long integration times

Authenticate the navigation message, e.g. via TESLA

- ▶ **not** because we care a lot about the navigation message (can easily be retrieved via HTTPS),
- ▶ **but** because it adds lots of unpredictable data bits.
- ▶ **can** be useful where receiver has accurate independent time, to verify timely ($1\ \mu\text{s}$) arrival of leading edge of random bits

Conclusions II

Random data bits are no substitute for the steganographic solution:

- ▶ useless to receivers without accurate ($1\ \mu\text{s}$) clock, as each data bit is fully revealed within a few tens of microseconds.

Steganographic spreading code could be added on top of normal signal in-phase at $< -30\ \text{dB}$ relative power

- ▶ does not have to support acquisition and tracking: receiver only needs to verify its presence to confirm timing of regular signal
- ▶ can be detected with long integration time (seconds)
- ▶ does not have to be continuously available, just serves to confirm samples of the navigation solution derived from regular signals.
- ▶ backwards compatible

References

- ▶ Logan Scott: Anti-spoofing and authenticated signal architectures for civil navigation signals. Proceedings ION GPS/GNSS 2003, pp. 1543–1552.
- ▶ Markus G. Kuhn: An asymmetric security mechanism for navigation signals, 6th Information Hiding Workshop, LNCS 3200, Springer-Verlag, 2004.
- ▶ Chris Wullems, Oscar Pozzobon, Kurt Kubik: Trust your receiver? Enhancing location security. GPS World, Oct 1, 2004.
- ▶ Oscar Pozzobon, Chris Wullems, Kurt Kubik: Secure tracking using trusted GNSS receivers and Galileo authentication services. Journal of Global Positioning Systems, Vol. 3, No. 1–2, pp. 200–207, 2004.
- ▶ Markus G. Kuhn: Signal authentication in trusted satellite navigation receivers. In Ahmad-Reza Sadeghi, David Naccache (Eds.): Towards Hardware-Intrinsic Security, Springer, 2011, ISBN 978-3-642-14451-6.
- ▶ Kyle Wesson, Mark Rothlisberger, Todd Humphreys: Practical cryptographic civil GPS signal authentication, NAVIGATION, Journal of the Institute of Navigation, Vol 59, Num 3, 2012.
- ▶ Todd Humphreys: Detection strategy for cryptographic GNSS anti-spoofing, IEEE Trans. Aerospace and Electr. Syst. 49(2), 2013.

<http://www.cl.cam.ac.uk/research/security/>