Distance-bounding protocols

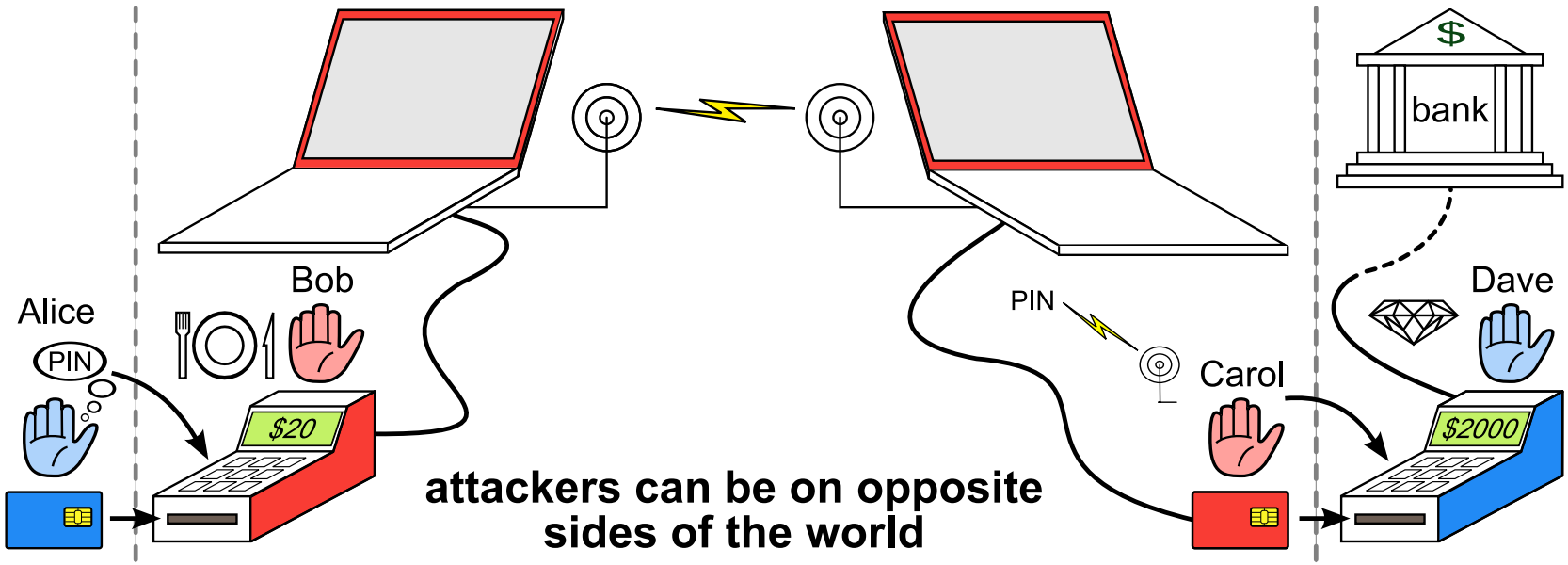
Markus Kuhn

with Gerhard Hancke, Saar Drimer, Steven Murdoch, et al.

UNIVERSITY OF
CAMBRIDGE
Computer Laboratory

https://www.cl.cam.ac.uk/~mgk25/

Relay attacks



[Drimer/Murdoch 2007]

- 2008 demonstration of EMV Chip&PIN relay attack on BBC TV
- card terminals tolerated delays of many seconds
- concerns about contact-less EMV transaction (no PIN up to £30)

www.dailymail.co.uk/news/article-5571407/The-rise-contactless-led-relentless-rise-co

Mail Online

'Tap-and-go' bank cards are so very convenient but read how fraudsters can steal your details just by standing close to you — and ask yourself... will you ever use contactless again?

- **Contactless has become a default for nearly all debit and credit cards in Britain**
- **But the rise of contactless has also led to the relentless rise of contactless fraud**
- **Some phones can double as card reader via app called 'credit card reader NFC'**
- **This technology allows bank cards to be skimmed through clothes and wallets**

By HARRY WALLOP FOR THE DAILY MAIL
PUBLISHED: 00:52, 3 April 2018 | **UPDATED:** 09:22, 3 April 2018

Distance-bounding protocols

- cryptographic challenge-response authentication protocol
- designed to provide strong upper bound for distance to proofer
- tight bounds (metres) difficult over regular data-communication channels (length of single bit, variability in bitrates, packet latency, headers and checksum trailers)

Applications:

- card-present payment transactions
- RFID door access control
- desktop authentication
- road-toll OBU
- military friend-foe identification
- prisoner tagging
- wireless sensor network security (wormhole routing attacks)

Location-finding techniques

Received Signal Strength (RSS): Uses the inverse relationship between signal strength and distance to estimate the distance to other nodes.

- But attacker can alter received signal strength: amplifier, higher-gain antenna, relay transponder, etc.

Angle-of-Arrival (AoA): Examines the directions of received signals to determine the locations of transmitters or receivers.

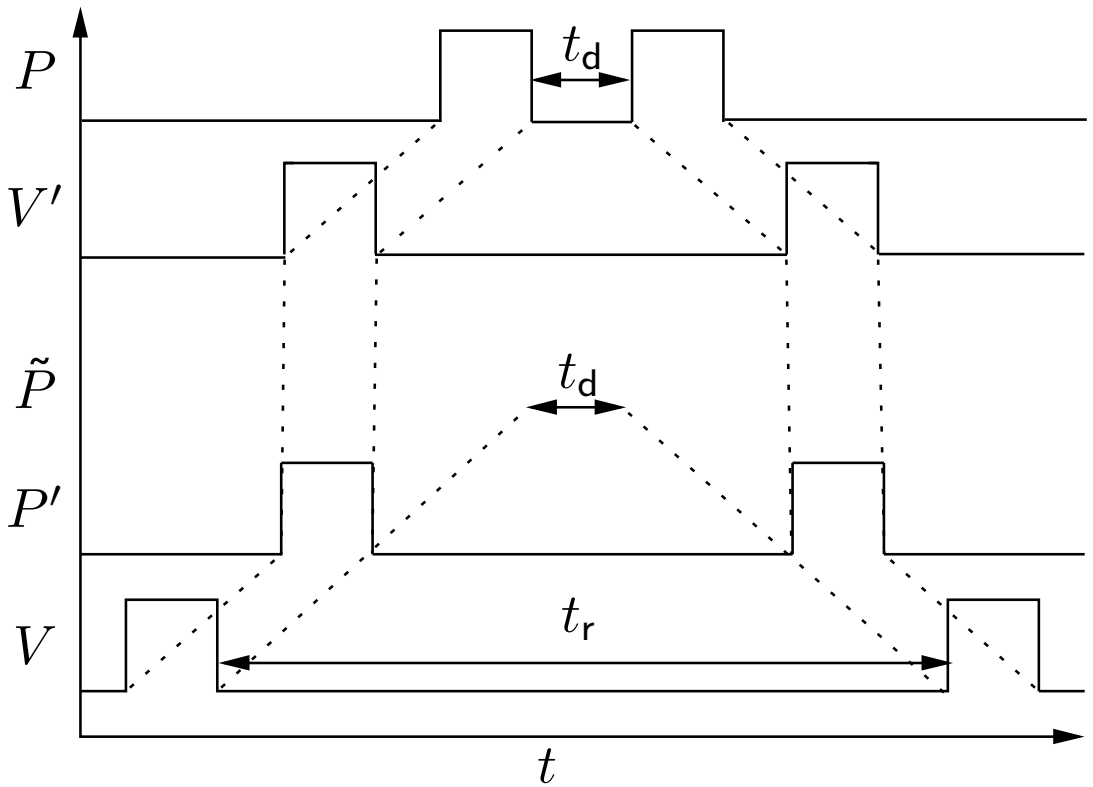
- But attacker can reflect/retransmit from a different direction.

Time-of-Flight (ToF): Measures elapsed time for a message exchange to estimate distance based on the communication medium's propagation speed.

- General Relativity: Universe does not propagate information faster than 30 cm/ns $=$ 300 m/μs $=$ 300 km/ms $= 3 \times 10^8$ m/s.
- Method of choice for high-security distance-bounding approaches.
- Practical near speed-of-light channels: contact, NFC, radio, optical

Acoustic/ultrasonic signals can be relayed via radio

Chose medium with propagation speed c close to speed of light.
Otherwise:



The vertical axis represents position. In this relaying attack, an attacker places a fake prover P' and a fake verifier V' near the actual verifier V and prover P, respectively. The exchanged data is related between P' and V' via a fast radio link. The shortened round-trip time t_r makes V believe that P is at the nearer position $\tilde{P}$.

Naïve approaches

Distance-bounding protocols: adapted authentication protocols to establish an upper bound for the distance of a prover P to a verifier V.

First attempt: a normal authentication protocol with a tight timing constraint:

$$\begin{aligned} V_{t_1} \to P_{t_2} : &\qquad C \in_{\mathsf{r}} \{0,1\}^n \\ P_{t_3} \to V_{t_4} : &\qquad R = \mathsf{Mac}_K(C) \end{aligned}$$

The distance bound is then

$$d(P,V) \leq \frac{t_{\mathsf{r}} - t_{\mathsf{d}}}{2c} = \frac{(t_4 - t_1) - (t_3 - t_2)}{2c}$$

where c is the signal propagation speed, t_{r} is the challenge-response round-trip time, and t_{d} is the processing delay in the prover P.

Problems with regular challenge-response protocols

- cryptographic functions (e.g., MAC) can take thousands of clock cycles to compute
- their inputs and outputs can take hundreds of clock cycles to transmit
- 10 000 clock cycles at 10 MHz $=$ 1 ms
- Basic crystal oscillator – 10^{-4} (100 ppm) relative frequency error
  1 ms $\pm$ 100 ns means $\pm$15 m distance error
- Internal RC oscillator – 10^{-1} relative frequency error
  1 ms $\pm$ 0.1 ms means $\pm$15 km distance error

  Contactless smartcards usually lack crystal oscillators and rely on terminal to provide time reference (RF carrier) to clock communication; may use internal RC circuit and LC tuning of antenna to bound clock frequency.

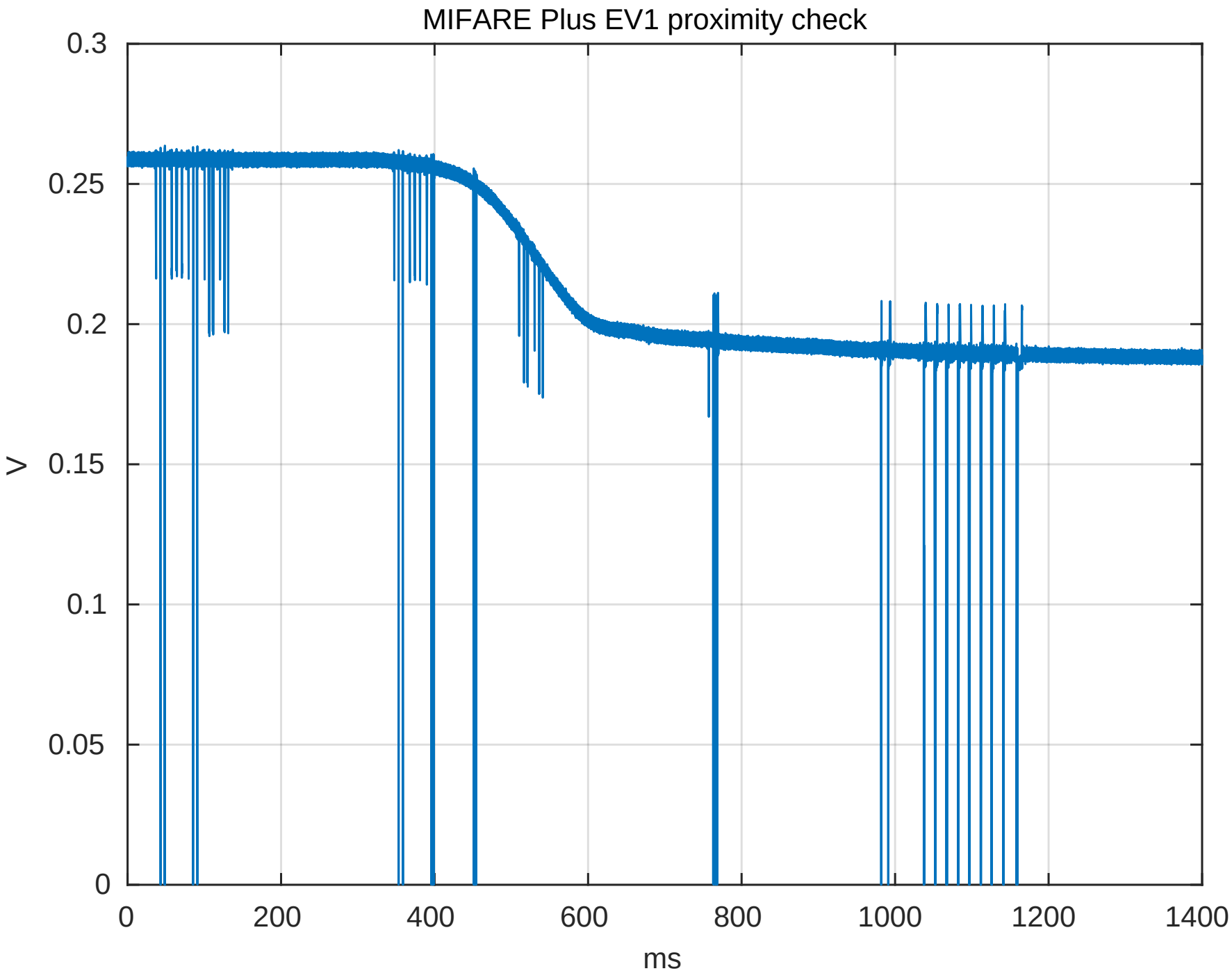
Trusted prover with trusted sampling clock

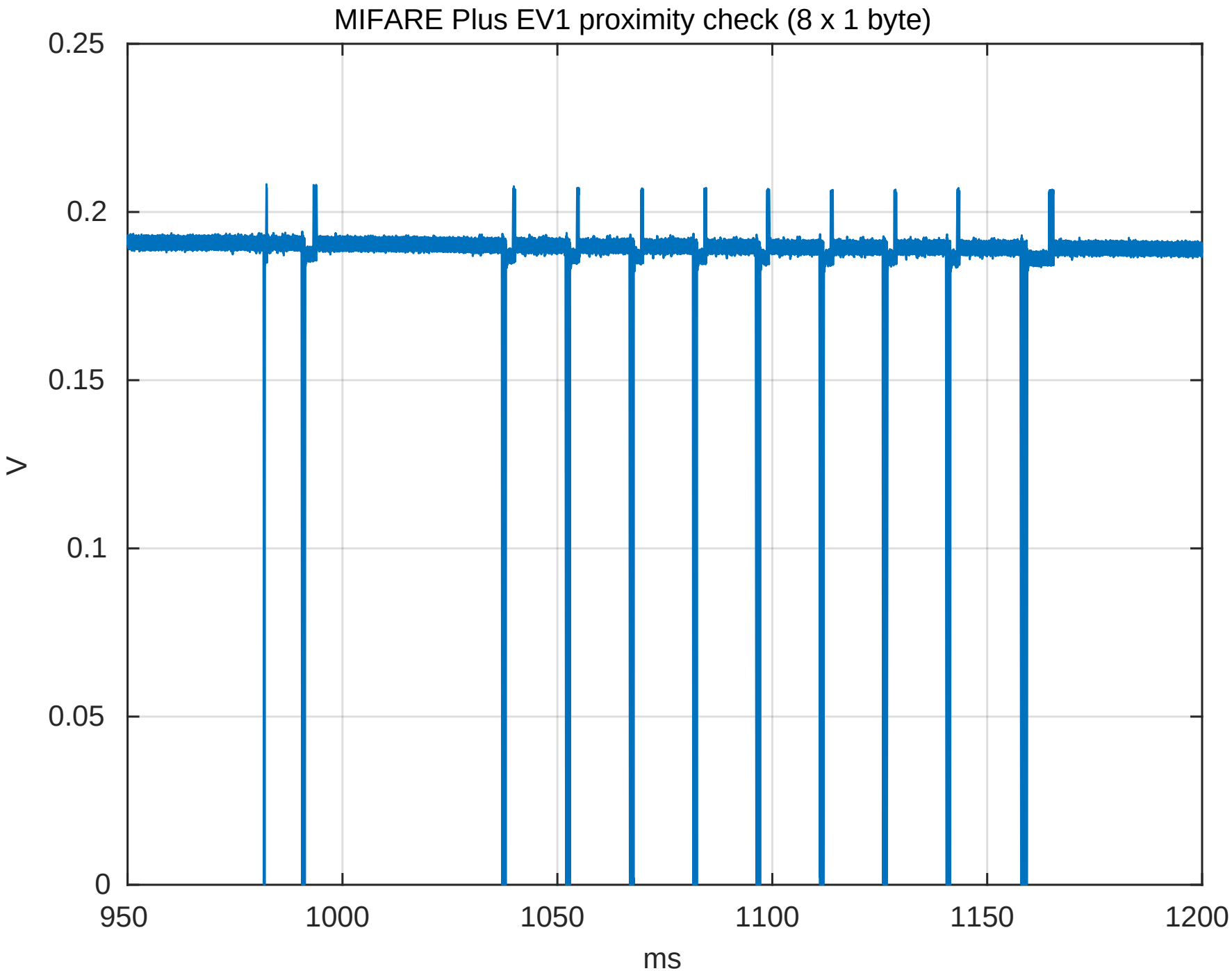
If a prover P is completely trusted (tamper-resistant hardware, tamper-resistant and stable reference clock):

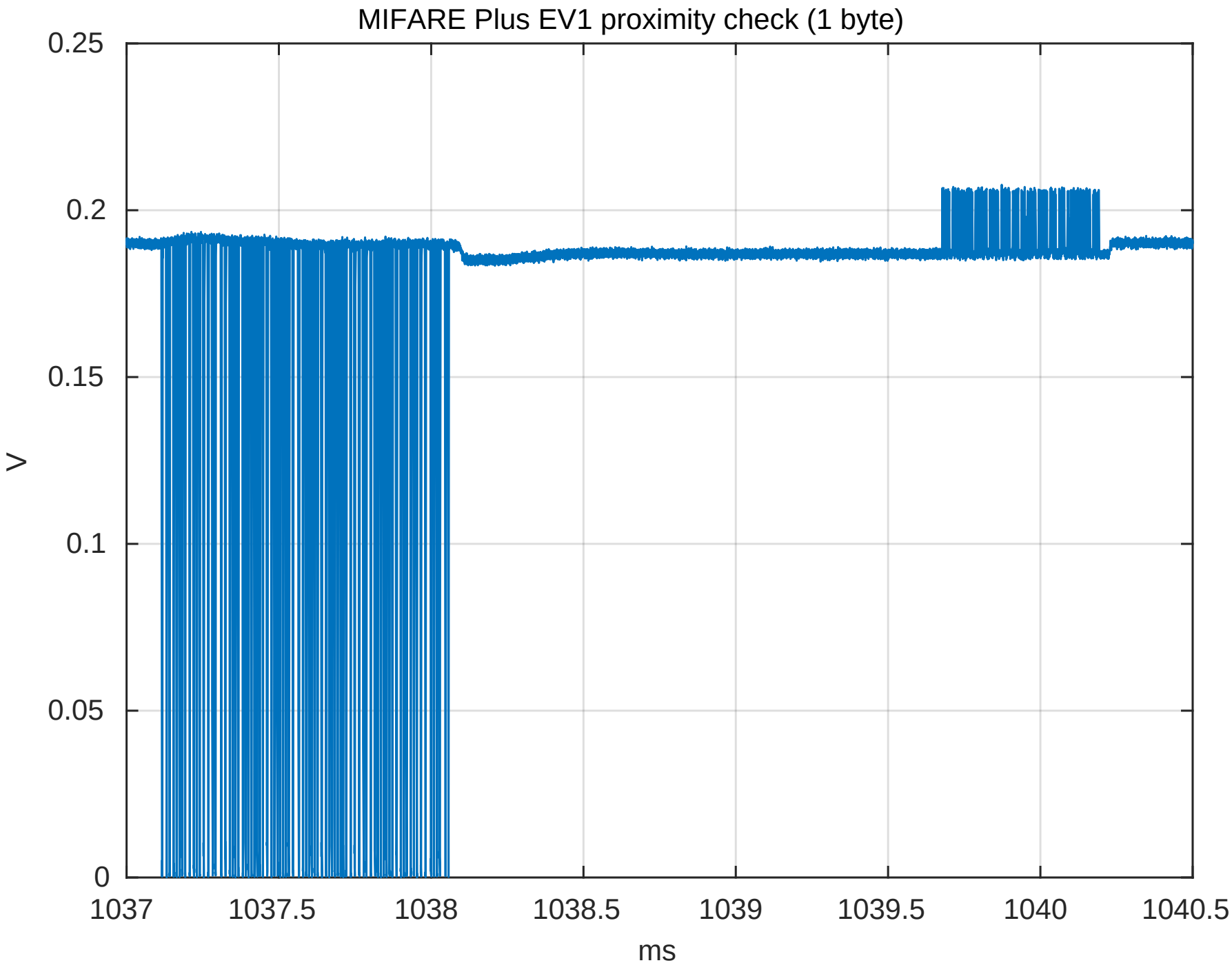
- V generates n random bits $C = C_1 C_2 \ldots C_n$
- P generates n random bits $R = R_1 R_2 \ldots R_n$
- P and V exchange parameters: $(t_R - t_C, n, \Delta t)$
- V sends C
- P samples incoming random bits $C = C_1 C_2 \ldots C_n$ at times $t_{i,C} = t_C + i \cdot \Delta t$ $(i \in \{1, \ldots, n\})$
- P sends its own random bits $R = R_1 R_2 \ldots R_n$ at times $t_{i,R} = t_R + i \cdot \Delta t$ $(i \in \{1, \ldots, n\})$
- P confirms to V the exchanged data afterwards (not time critical): $P \to V : \mathrm{Mac}_K(t_R - t_C, n, \Delta t, C, R)$

With relative frequency error η in P's clock, the resulting distance uncertainty is approximately $\eta c |t_R - t_C|$. Therefore keep $|t_R - t_C|$ as small as possible: $|t_R - t_C| \approx 0$ if duplex transmission is available and $|t_R - t_C| \approx \Delta t(n+1)$ on half-duplex channels.

Commercial implementation: MIFARE Plus/DESFire Proximity Check


MIFARE Plus EV1 proximity check
0.3
0.25
0.2
0.15
0.1
0.05
0
V
0
200
400
600
800
1000
1200
1400
ms


MIFARE Plus EV1 proximity check (8 x 1 byte)
0.25
0.2
0.15
0.1
0.05
0
V
950
1000
1050
1100
1150
1200
ms


MIFARE Plus EV1 proximity check (1 byte)
0.25
0.2
0.15
0.1
0.05
0
V
1037
1037.5
1038
1038.5
1039
1039.5
1040
1040.5
ms

MIFARE Proximity Check

MIFARE Plus EV1 and DESFire EV2 RFID cards support now a basic distance-bounding protocol, running on top of the same standard ISO 14443-4 ("T=CL") protocol and ISO 7816-4 APDU format used for other card transactions.

- RF channel: 13.56 ± 2 MHz (ISO 14443 Type A)
- reader to card: 848 kbit/s, 1 bit $= 4.72$ µs
- card to reader: 212 kbit/s, 1 bit $= 1.18$ µs
- C and R are 8-bytes long each, can be sent all at once or split aross up to eight 1-byte APDUs
- header and CRC overhead due to T=CL and APDU wrapping
- challenge packet: 0.94 ms, response packet: 0.52 ms
- intra-packet gap: 1.7 ms promised, 1.6 ms actual
- 0.1–0.2 ms slack $\rightarrow$ 15 km distance uncertainty
- Linux protocol stack (USB CCID, PCSC-Lite, Perl Chipcard::PCSC::Card) round-trip latency: 14.67 ± 0.1 ms
  $\Rightarrow$ hardware timestamping support in reader desirable

```
reader = ACS ACR122U PICC Interface 00 00
->card 90 f0 00 00 00
<-card 01 06 a0 07 91 90
PubRespTime: 1696 µs
PPS1: 07
DS: 2, card->reader 211.875 kbit/s, etu=4.71976 µs
DR: 8, reader->card 847.5   kbit/s, etu=1.17994 µs
->card 90 f2 00 00 02 01 6c 00
<-card 4b 91 90
elapsed time: 14.632 ms
->card 90 f2 00 00 02 01 c1 00
<-card 78 91 90
elapsed time: 14.772 ms
->card 90 f2 00 00 02 01 8b 00
<-card b8 91 90
elapsed time: 14.652 ms
->card 90 f2 00 00 02 01 d6 00
<-card b9 91 90
elapsed time: 14.607 ms
->card 90 f2 00 00 02 01 0e 00
<-card fa 91 90
elapsed time: 14.7 ms
->card 90 f2 00 00 02 01 6d 00
<-card 11 91 90
elapsed time: 14.596 ms
->card 90 f2 00 00 02 01 f0 00
<-card 28 91 90
elapsed time: 14.593 ms
->card 90 f2 00 00 02 01 56 00
<-card 0d 91 90
elapsed time: 14.597 ms
mac input: fd 01 06 a0 07 4b 6c 78 c1 b8 8b b9 d6 fa 0e 11 6d 28 f0 0d 56
->card 90 fd 00 00 08 51 b2 50 b0 b8 8e c4 95 00
<-card 80 29 e5 4d 52 78 f2 28 91 90
```

EMV®
Contactless Specifications for Payment Systems

Book C-2

Kernel 2 Specification

Version 2.6
February 2016

3.10 Relay Resistance Protocol

3.10.1 Introduction

A relay attack is where a fraudulent terminal is used to mislead an unsuspecting cardholder into transacting, where the actual transaction is relayed via a fraudulent Card (or simulator) to the authentic terminal of an unsuspecting merchant. It may also be that a fraudulent reader is used without the cardholder being aware of the transaction.

3.10.2 Protocol

The relay resistance protocol works as follows:

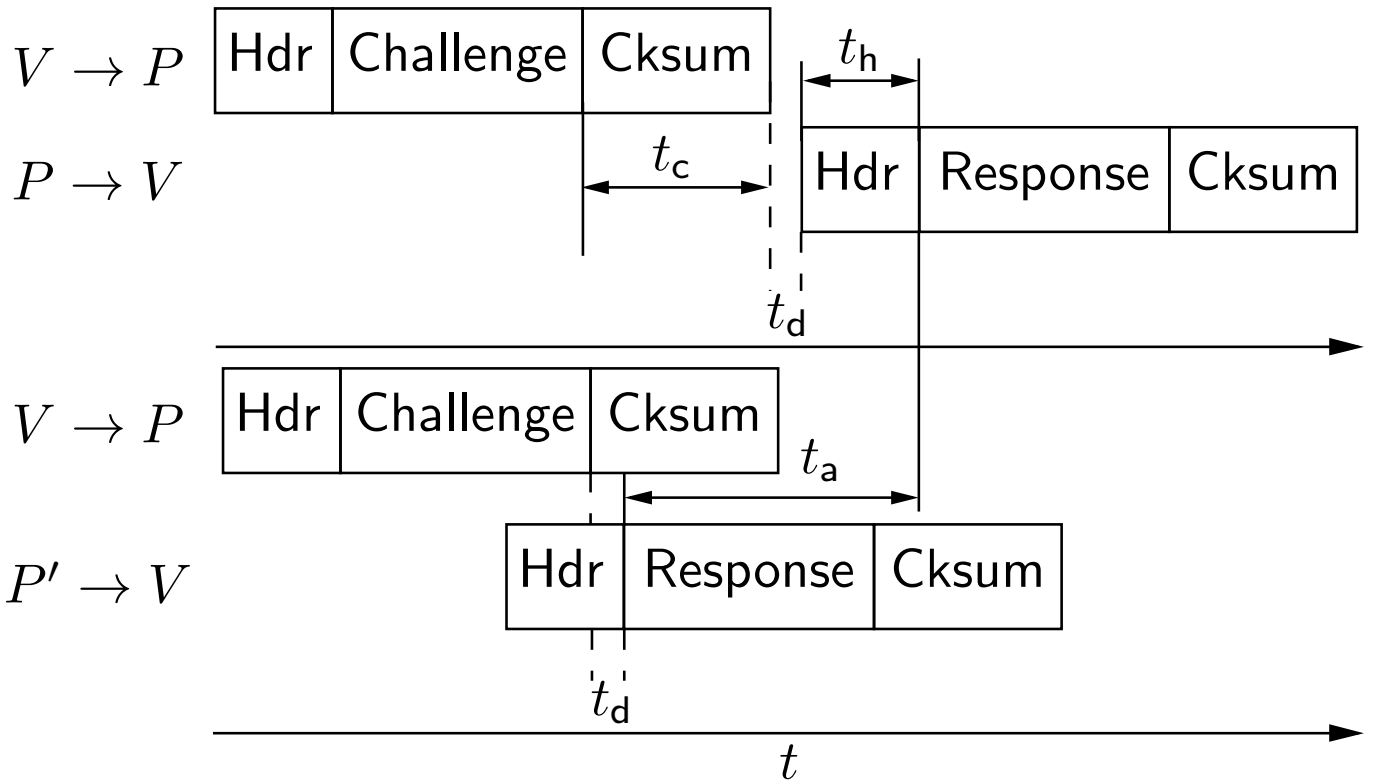
1. A bit in Application Interchange Profile is used to tell the Reader that the Card supports the relay resistance protocol. A bit in *Kernel Configuration* is used to configure the support of the relay resistance protocol by the Reader.
2. The Reader invokes the relay resistance protocol if both the Card and Reader support it. In this case it sends a timed C-APDU (EXCHANGE RELAY RESISTANCE DATA) to the Card with a random number (*Terminal Relay Resistance Entropy*). The Card responds with a random number (*Device Relay Resistance Entropy*) and timing estimates (*Min Time For Processing Relay Resistance APDU*, *Max Time For Processing Relay Resistance APDU* and *Device Estimated Transmission Time For Relay Resistance R-APDU*).
3. If the timings determined by the Reader exceed the maximum limit computed, the Reader will try again in case there was a communication error or in case other processing on the device interrupted the EXCHANGE RELAY RESISTANCE DATA command processing. The Reader will execute up to two retries.
4. *Terminal Verification Results* are used to permit the Reader to be configured

MasterCard Relay Resistance Protocol

- specified in EMV Contactless Specifications for Payment Systems, Book C-2, Kernel 2 Specification, Version 2.6, February 2016
- timed exchange of two 32-bit random numbers + metadata
- integrated with EMV's existing "Combined Dynamic Data Authentication and Application Cryptogram Generation" (CDA) protocol
- lots of timing parameters provided by both card and terminal, not just maximum but also minimum values (0.1 ms resolution):
  - Min Time For Processing Relay Resistance APDU
  - Max Time For Processing Relay Resistance APDU
  - Device Estimated Transmission Time For Relay Resistance R-APDU
  - Minimum Relay Resistance Grace Period
  - Maximum Relay Resistance Grace Period
  - Relay Resistance Accuracy Threshold
  - Relay Resistance Transmission Time Mismatch Threshold
- specification lacks clear definitions of when to start and stop the timer (in relation to bit edges as seen on a 13.56 MHz AM demodulated ISO 14443-3 A/B channel)

Protocol headers permit low-latency bypass

If packet processing happens outside the trusted hardware module (e.g., TPM does only the MAC) and the trusted hardware interface is much faster than the communications channel used:



Normal communication hardware requires software to commit to the full data packet some time before the first bit is actually sent, and notifies the software some time after the last bit is received.

An attacker can use special hardware without these restrictions.

Low-latency relaying hardware

Fastest relay strategies: analog up-down conversion of carrier or direct modulation of baseband signal (AM, OOK, BSK, FM, FSK):

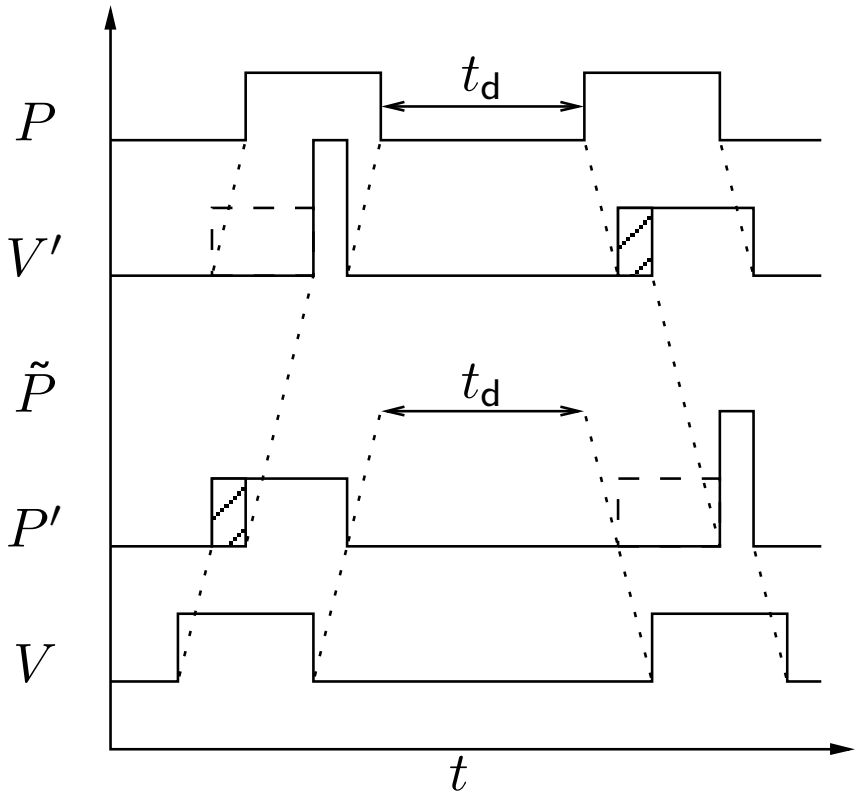
- Processing latency is dominated by the group delay (impulse response duration) of the analog filters involved, approximately 1/bandwidth, or $\approx 1 \ldots 5$ symbols.
- Without repeaters range of mobile UHF transmitters practically limited to a few kilometers (much more if airborn).

Various proprietary IoT UHF transceiver chips (e.g., Murata TRC103) support a low-latency "continuous mode" that bypasses packet buffering, passes each bit directly to the modulator, and thereby limits processing latency to a few tens of microseconds. Typical bitrates: 200–250 kbit/s.

Other transceivers (e.g., TI CC2420) offer pin signal when start-of-frame delimiter is detected. Still data packet latency, but accurate timing of packet arrival time. [Sommer, 2011]

Common standard digital wireless protocol implementations (802.11, Bluetooth, LTE, etc.) all add latencies of at least a few milliseconds, often more than 10 ms.

Special modulator delays commitment on bit value

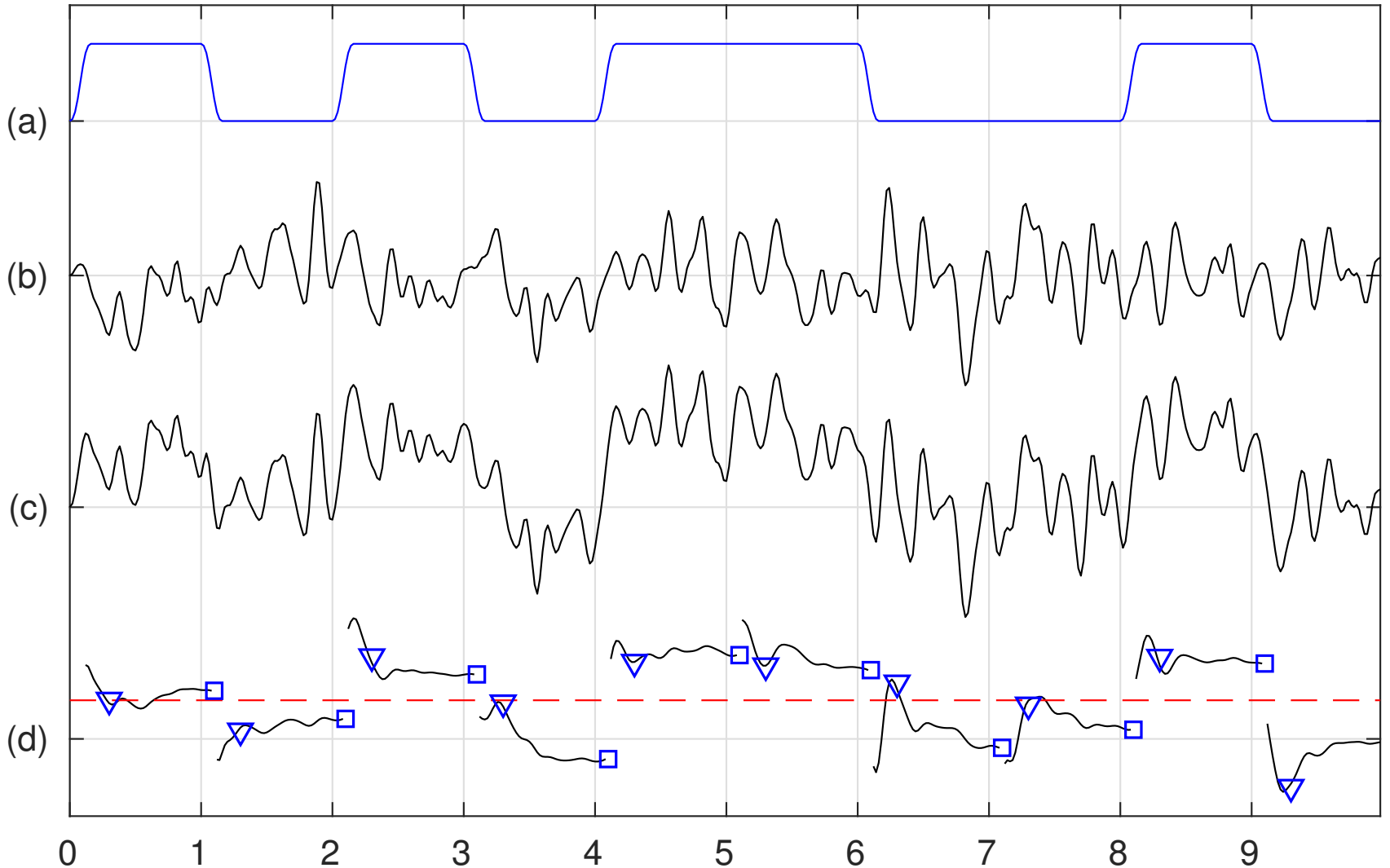


Standard symbol detectors integrate the signal received during the timeslot allocated to a bit, before deciding whether the total energy received was above or below the 0/1 decision threshold (matched filter).

An attacker can place the symbol's energy at the end of the bit slot and can decide on a bit value near the beginning of the slot, thereby bypassing some latency.

At 300 kbit/s (faster than most RFID protocols), a bit is 1 km long.

Special demodulator provides early bit estimate



(a) transmitted signal, (b) channel noise, (c) received signal, (d) integrator output in detector

Secure distance-bounding protocols

Principle 1: Use a communication medium with a propagation speed as close as possible to the physical limit for propagating information: the speed of light in vacuum.

This excludes not only acoustic communication techniques, but also limits applicability of wires and optical fibers.

Principle 2: Use a communication format in which only a single bit is transmitted and the recipient can instantly react on its reception.

This excludes most traditional byte- or block-based communication formats, and in particular any form of error correction.

Principle 3: Minimize symbol length.

In other words, output the energy that distinguishes the two possible transmitted bit values within as short a time as is feasible. This leaves the attacker little room to shorten this time interval further.

Principle 4: The protocol should cope well with substantial bit error rates during the rapid single-bit exchange.

Principle 3 limits the energy that can be spent on transmitting a single bit and conventional error correction is not applicable.

Secure distance-bounding protocols

If timing of communications channel may be manipulated by attacker, ask prover to instantly evaluate a fast function for each challenge bit:

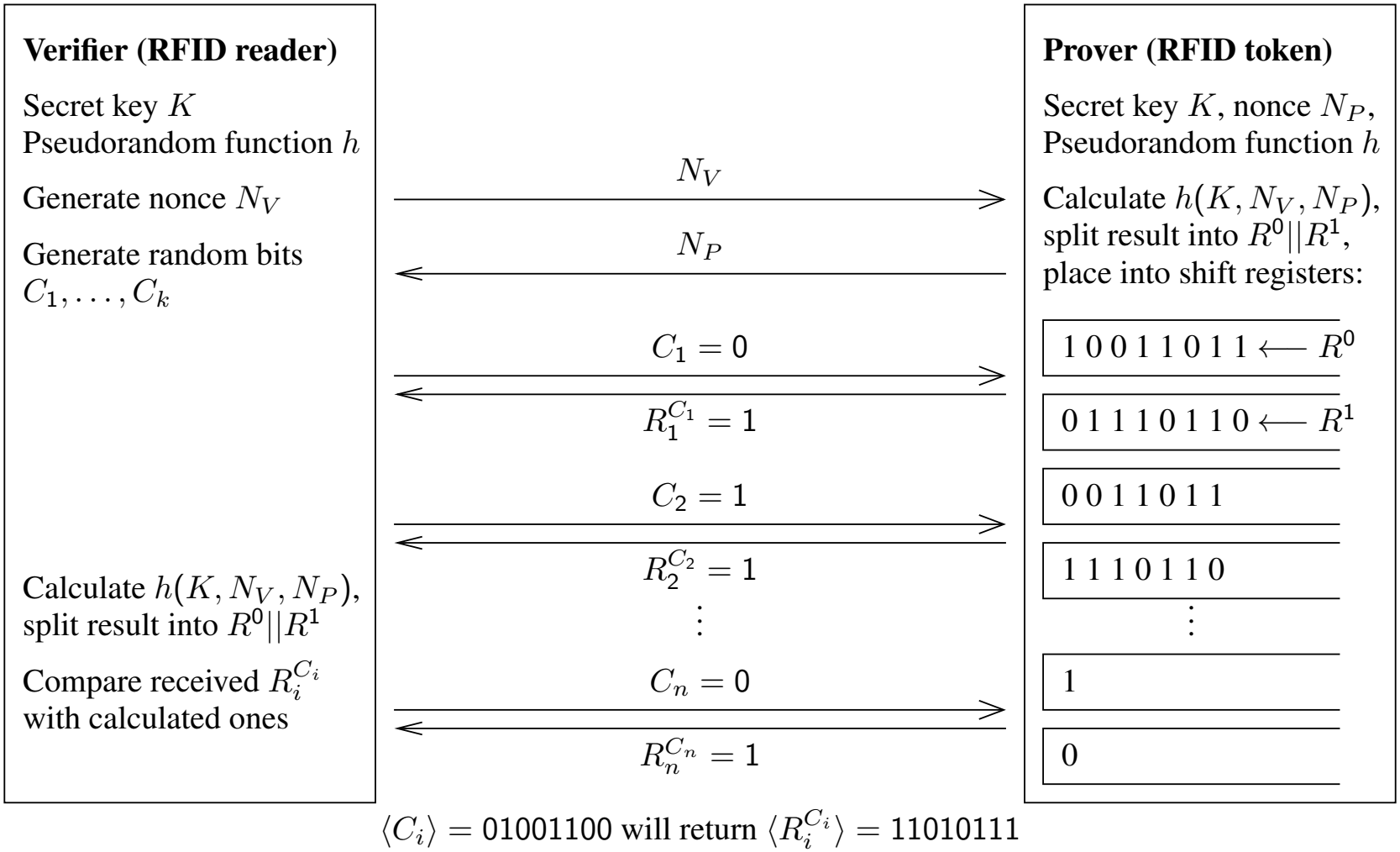
Brands/Chaum (1993): XOR

- V generates bit sequence C, P generates bit sequence M
- P commits to M.
- $\forall i : V$ sends bit C_i, P instantly answers $R_i = C_i \oplus M_i$
- P opens commitment on M and signs C.
- attacker can guess each M_i with 50% probability

Hancke/Kuhn (2005): 1-bit lookup

- avoids need for commitment and final signature, just needs some session-key setup (can be amortized with other transactions!)
- permits rapid/short bit exchange on noisy channel
- attacker can guess each R_i with 75% probability by guessing 50% of challenge bits correctly
- requires more rapid-response bits than Brands/Chaum for equal security, but saves other bits and can use faster bit rate due to error tolerance

Hancke/Kuhn protocol



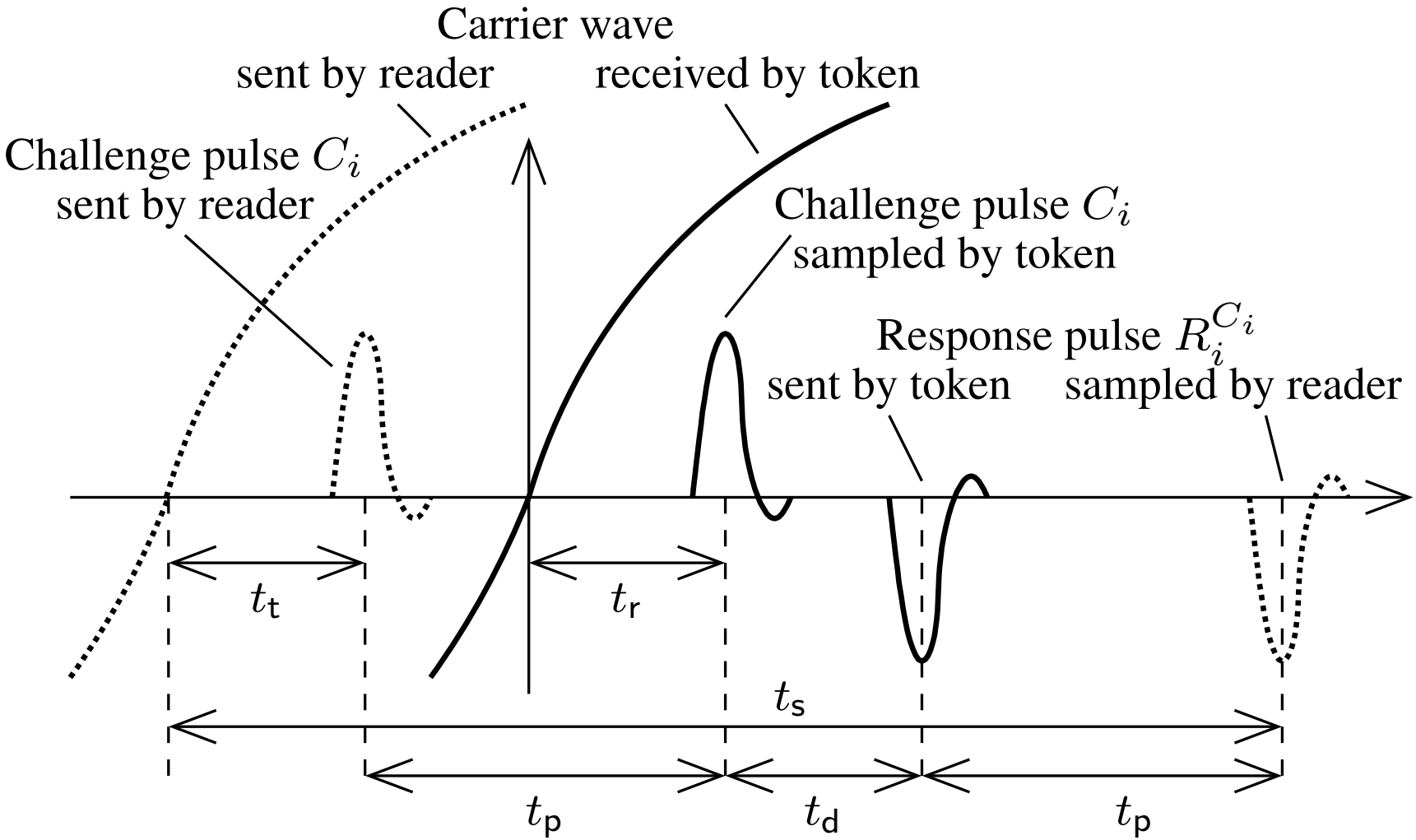
$\langle C_i \rangle = 01001100$ will return $\langle R_i^{C_i} \rangle = 11010111$

Security/robustness: V must get at least k of the n bits $R_i^{C_i}$ correctly.

False accept rate: $p_{\mathsf{FA}} = \sum_{i=k}^{n} \binom{n}{i} \cdot \left(\frac{3}{4}\right)^i \cdot \left(\frac{1}{4}\right)^{n-i}$

False reject rate: $p_{\mathsf{FR}} = \sum_{i=0}^{k-1} \binom{n}{i} \cdot (1-\epsilon)^i \cdot \epsilon^{n-i}$ (bit-error rate ϵ)

Ultra-wideband pulse communication (contactless)

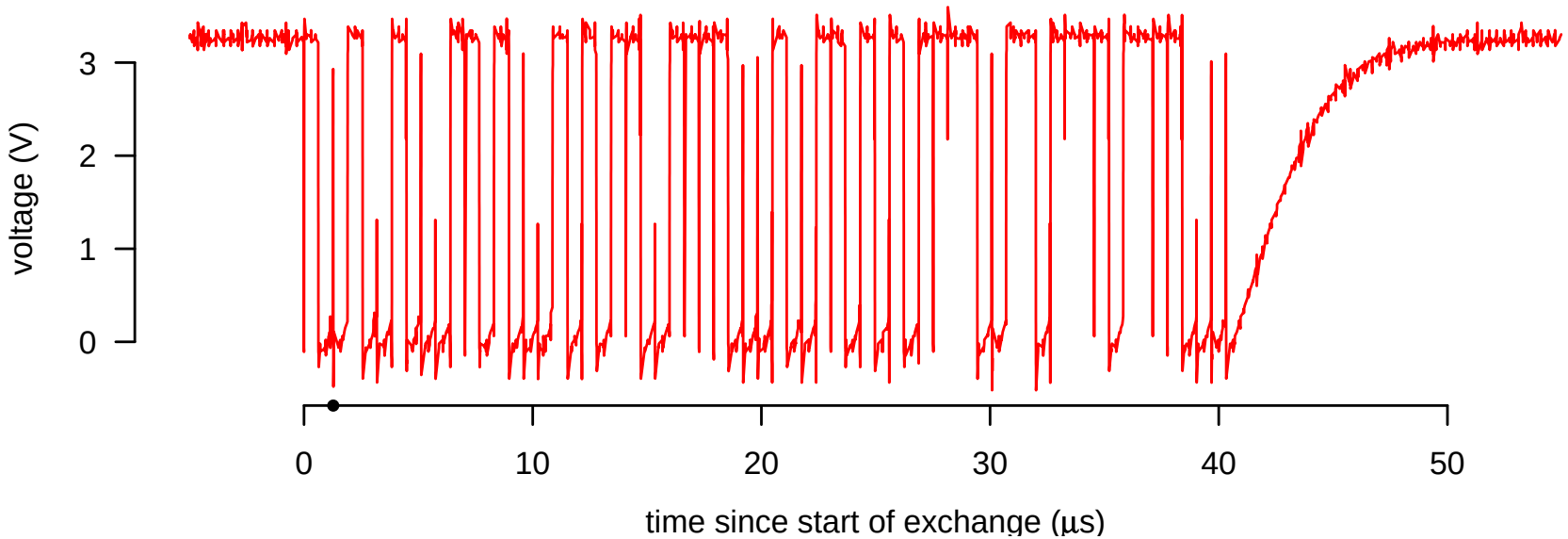


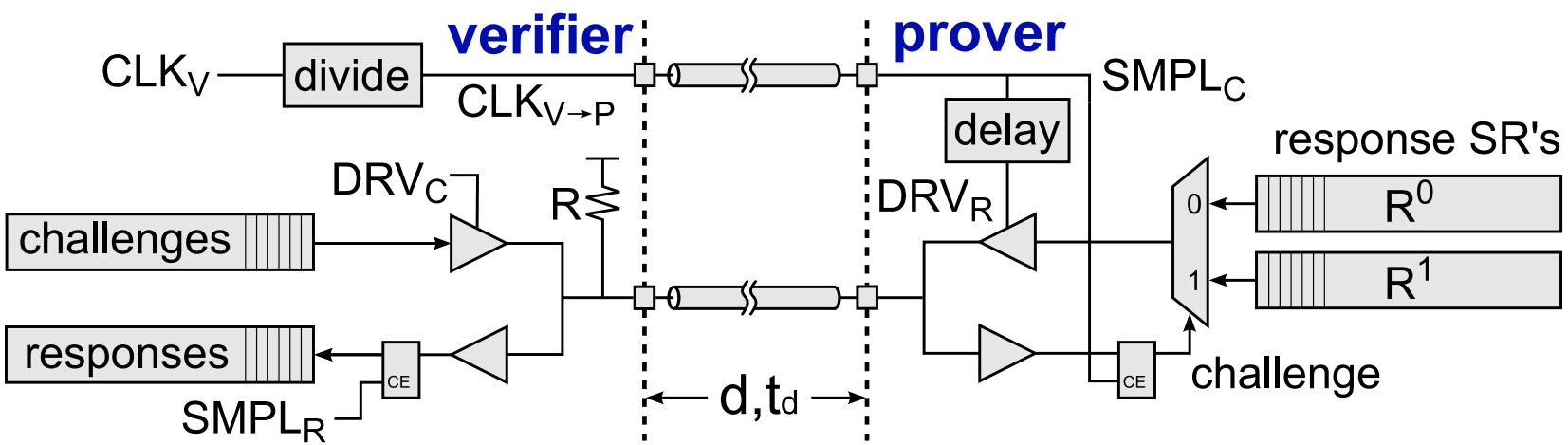
Asymmetric calibration requirement: prover can be low-cost token without trustworthy clock. Verifier (e.g., wall-mounted reader) performs a search to find the best t_t value to match circuit tolerances of t_r in prover.

Verifier may also search for best t_s to match t_d in prover chip and propagation delay t_p.

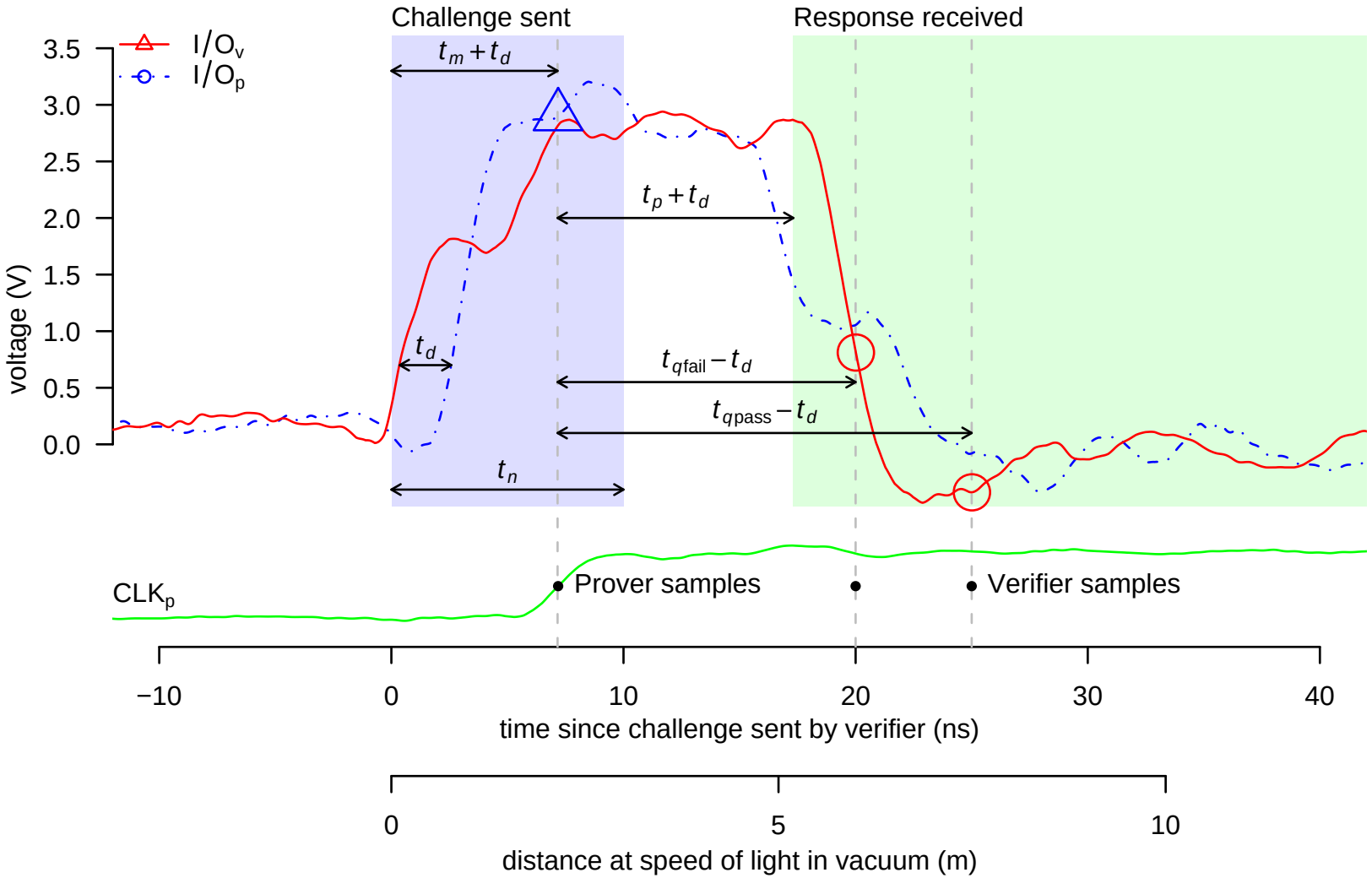
HK demonstration implementation (for ISO 7816 contacts)

Drimer/Murdoch 2007: Hancke/Kuhn over ISO 7816-style half-duplex contact interface with 25 ns roundtrip time (3.75 m abs. distance bound).

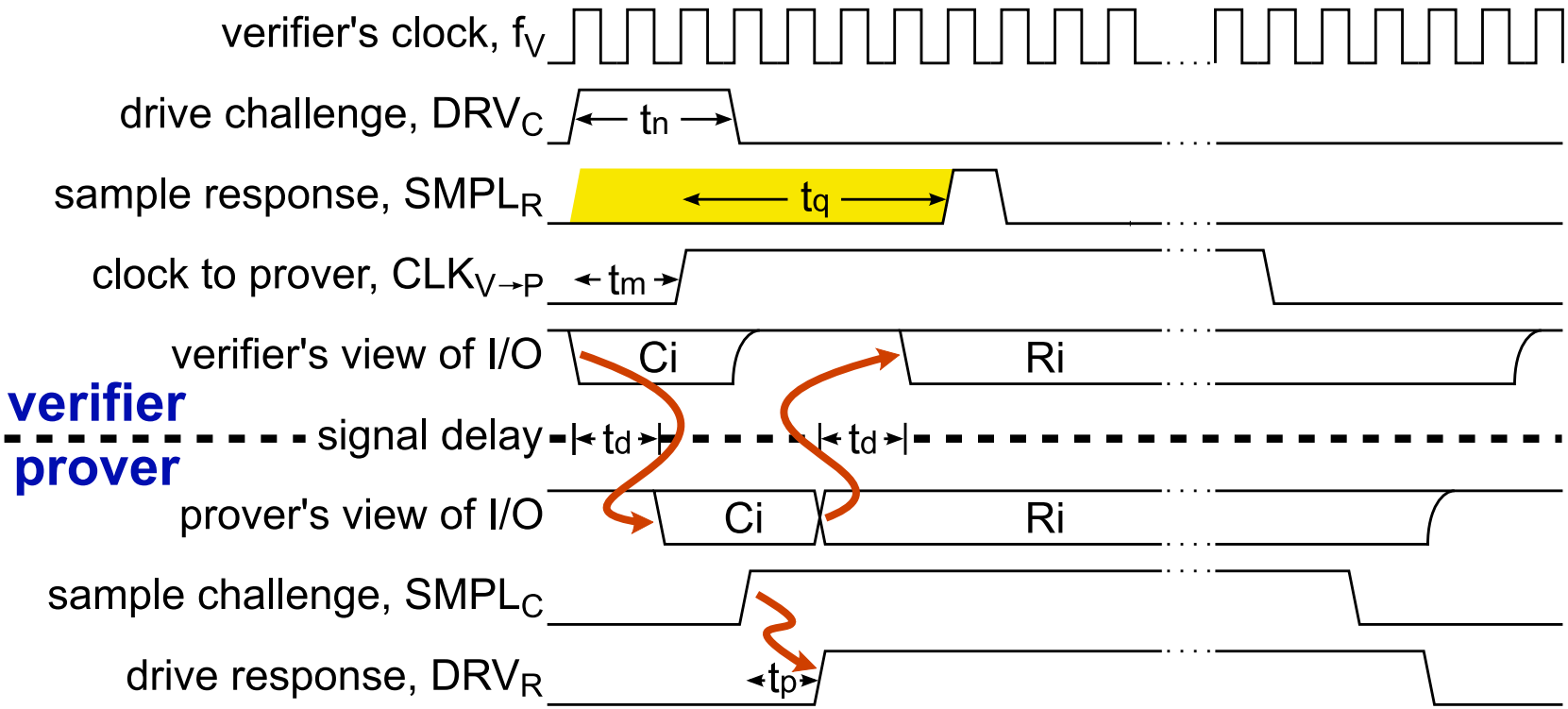




HK demonstration implementation (single-bit exchange)



HK demonstration implementation (control timing)



Conclusions

Tamper-resistant positioning services

- may be required in a wide range of applications
- have to take into account attacks with specialized hardware
- cannot easily be added later at the application protocol layer
- must be designed into the physical protocol layer
- rely on more than just tamper-resistant hardware modules
- require transmission and reception mechanisms that differ substantially from standard ones:
  - rapid single-bit round-trip exchanges for distance bounding
  - delayed correlation of weak signals for satellite positioning
- are another excellent example for an application where security must be considered in the design from the very beginning

References (distance bounding)

- Stefan Brands, David Chaum: Distance bounding protocols. Eurocrypt 1993, LNCS 765.
- Gerhard P. Hancke, Markus G. Kuhn: An RFID distance bounding protocol. IEEE SecureComm 2005, Athens, 2005, ISBN 0-7695-2369-2.
- Gerhard P. Hancke: Practical attacks on proximity identification systems. IEEE Symposium on Security and Privacy, Oakland, 2005.
- Jolyon Clulow, Gerhard P. Hancke, Markus G. Kuhn, Tyler Moore: So near and yet so far: distance-bounding attacks in wireless networks. European Workshop on Security and Privacy in Ad-Hoc and Sensor Networks, Hamburg, 2006, LNCS.
- Gerhard P. Hancke: A practical relay attack on ISO 14443 proximity cards, February 2005.
- Saar Drimer, Steven J. Murdoch: Keep your enemies close: Distance bounding against smartcard relay attacks. 16th USENIX Security Symposium, Boston, MA, August 2007.

`http://www.cl.cam.ac.uk/~mgk25/publications.html`
`http://www.cl.cam.ac.uk/research/security/`

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

Remote spoofing threat

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)

- Spoofer interferes from a distance (temporarily)
- Local antenna trusted, may see a mix of genuine and spoofed signals

Local spoofing threat

In some civilian applications: → Remote attestation

- Person in possession of GNSS receiver is not trusted
- Antenna manipulated, replaced, covered (long-term)
- Spoofer fully controls local RF-input port of trusted GNSS receiver

Local spoofing: existing real-word sensor attacks

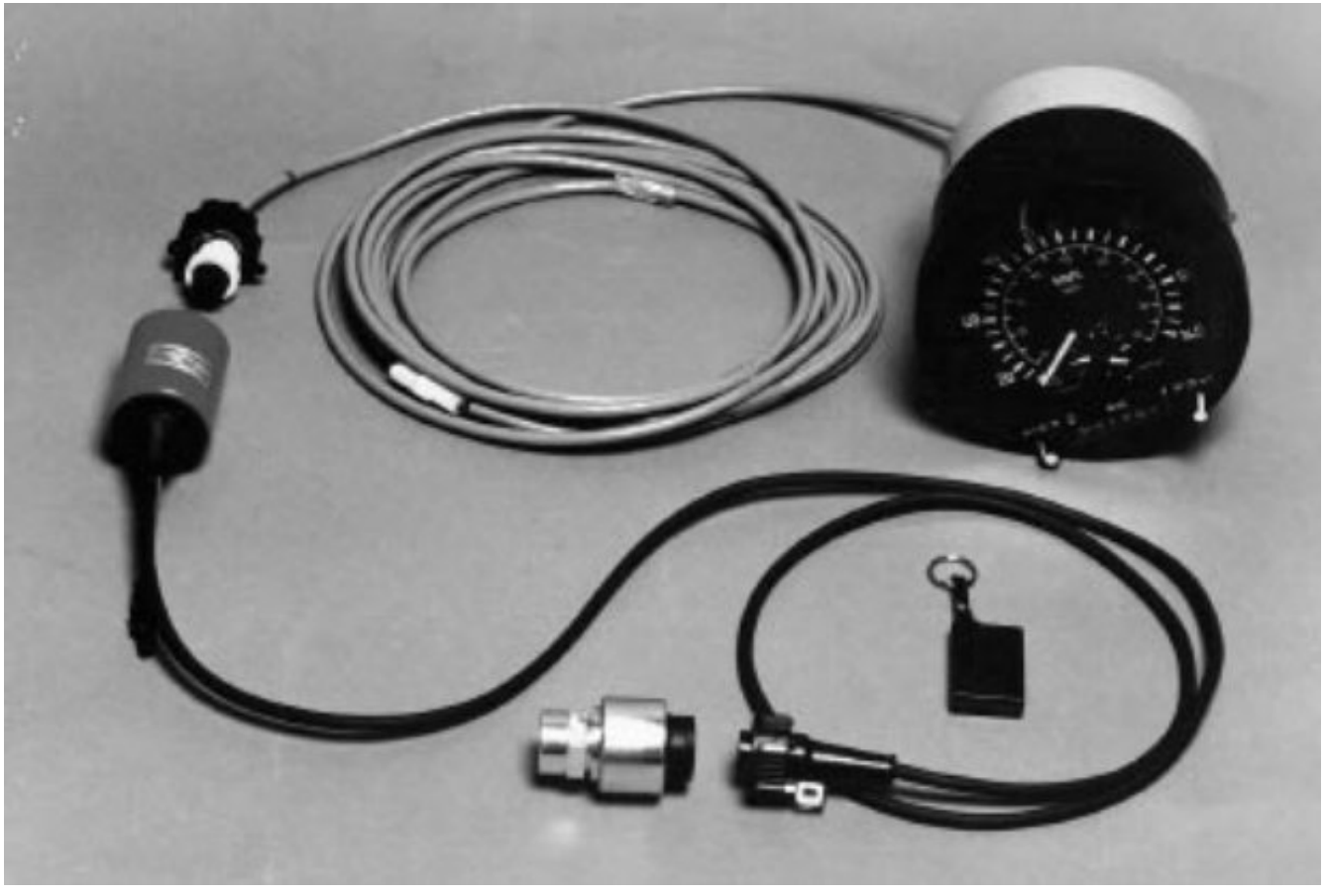
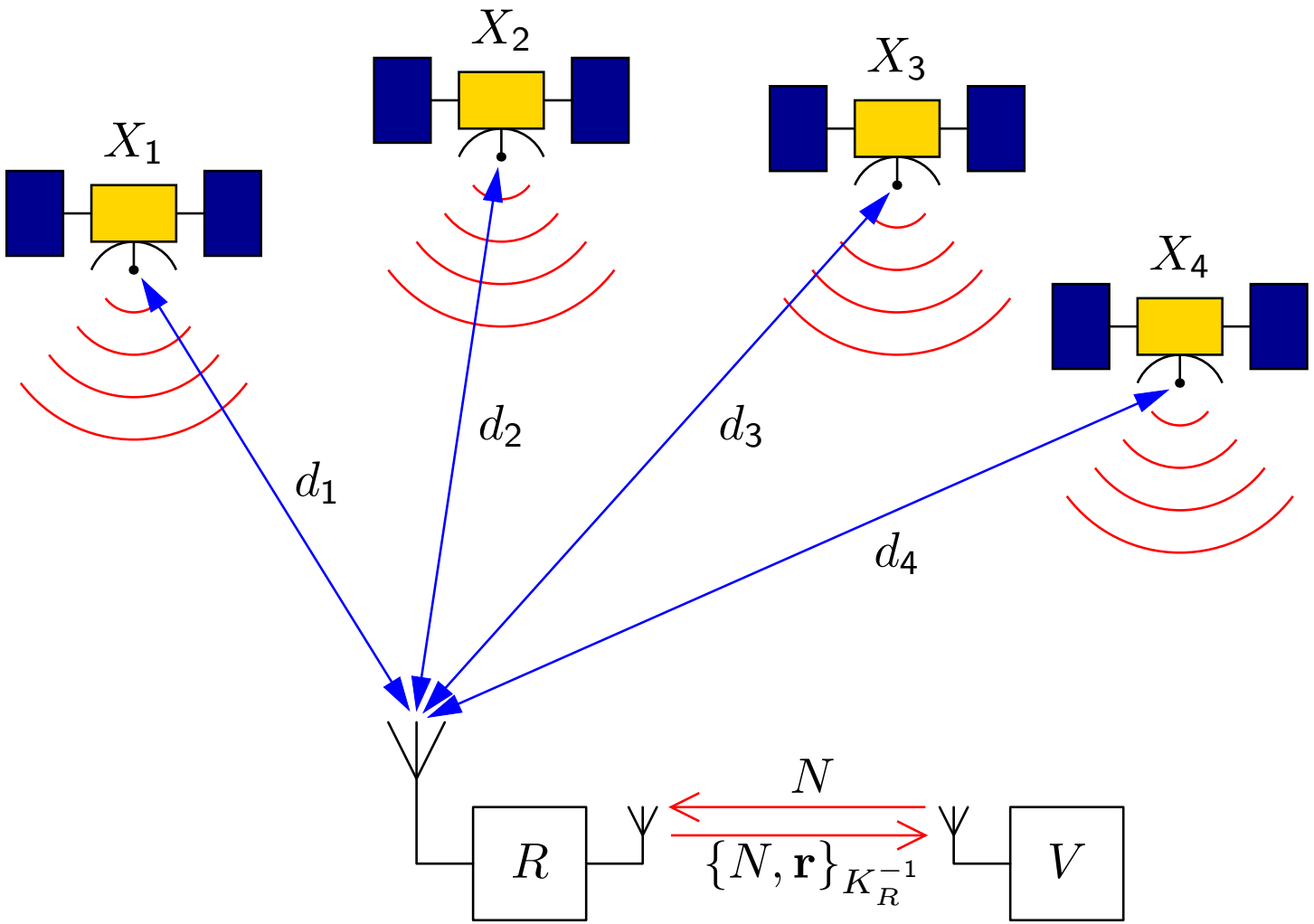


Photo: Hampshire Constabulary / Ross Anderson

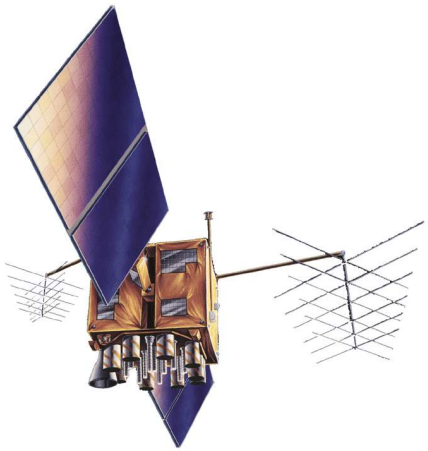
PVT-sensor spoofing devices have already been found "in the wild" by British police: in commercial good vehicles between tachograph and gearbox sensor. Drivers use them to manipulate their velocity and working-hours record.

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* $\mathbf{r}$.

Example application: offender tagging



GNSS signal

$\Downarrow$

Short range RF

$\Leftrightarrow$

GSM

$\Leftrightarrow$

RF bracelet

GNSS/GSM unit

Attacks on an offender tagging system

- sabotage unit and pretend malfunction
- detach RF bracelet without raising alarm
- relay between GNSS/GSM unit and distant GSM base station
- tamper with GNSS/GSM unit (extract keys, modify firmware) to spoof location-attestation protocol
- relay between bracelet and distant GNSS/GSM unit
- spoof GNSS signal, as it would be received elsewhere

Remote attestation

An offender tagging system is just one example of a GNSS application where the person in possession of the receiver has an interest in obtaining a fake navigation solution.

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, ...

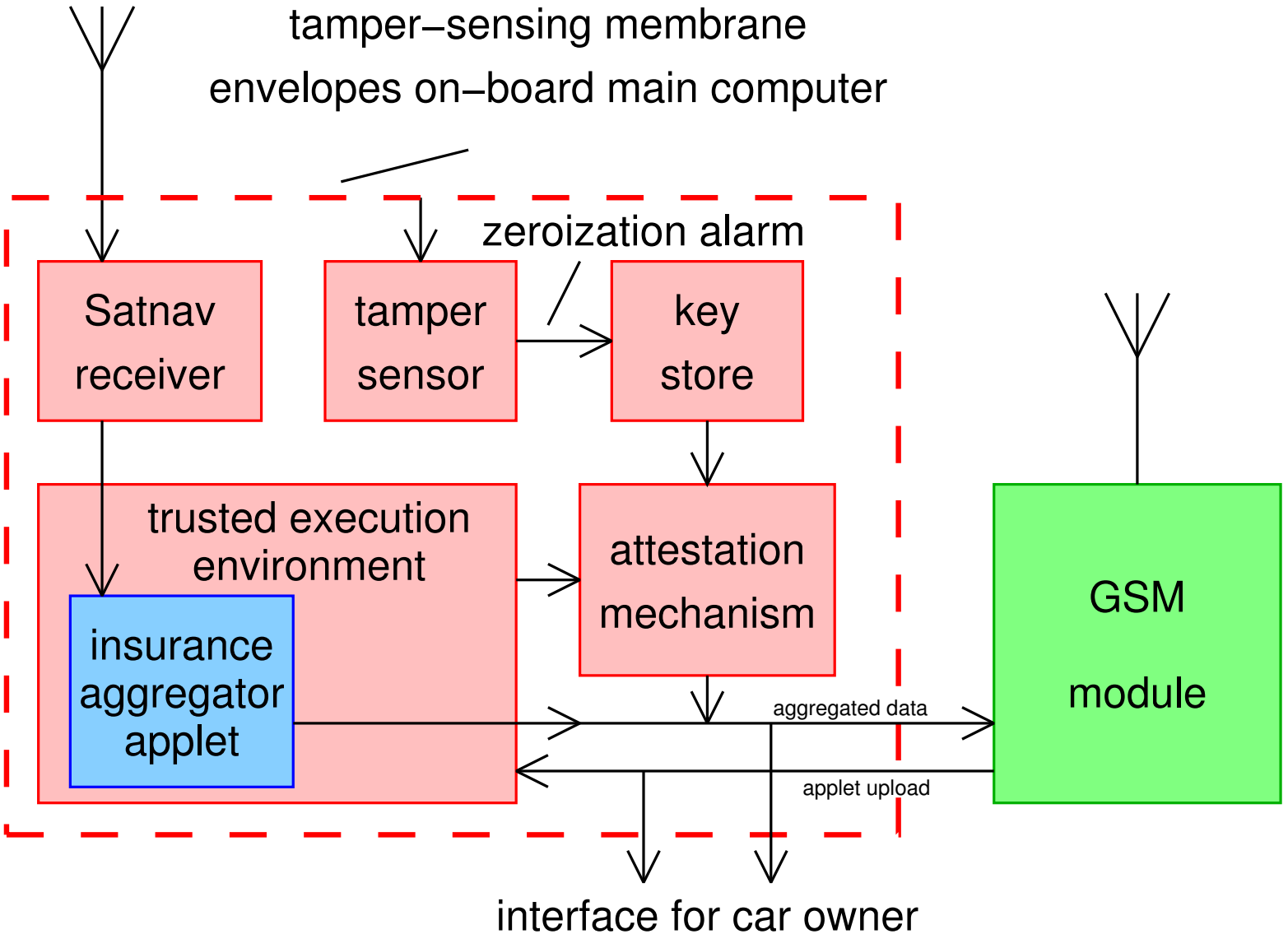
Potential legislative/contractual requirement, adversarial user, tamper-resistance requirement

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 $\rightarrow$ 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).

Secure positioning – cryptography at light speed

Conventional cryptographic authentication protocols

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

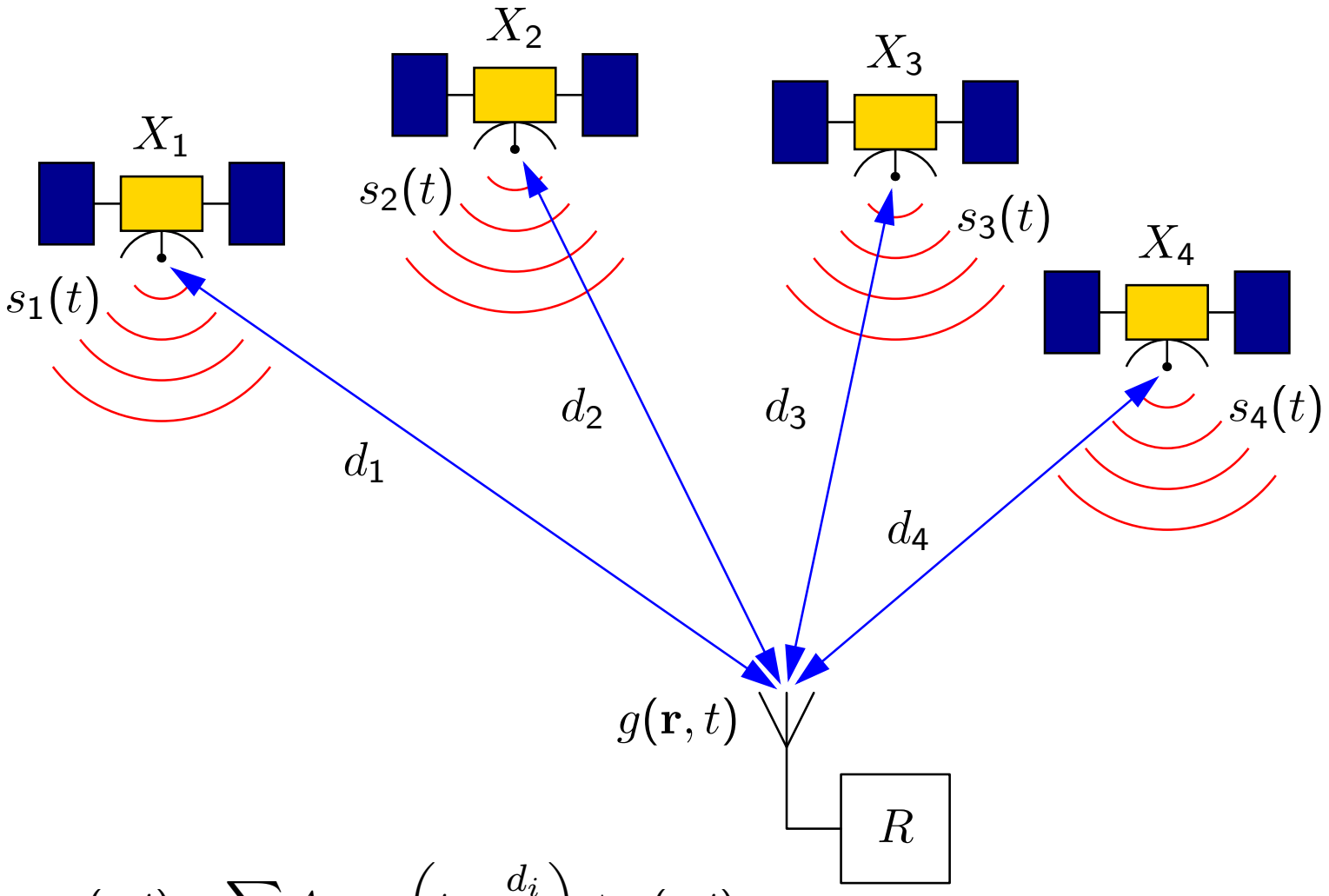
but do not authenticate

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

Protection technologies:

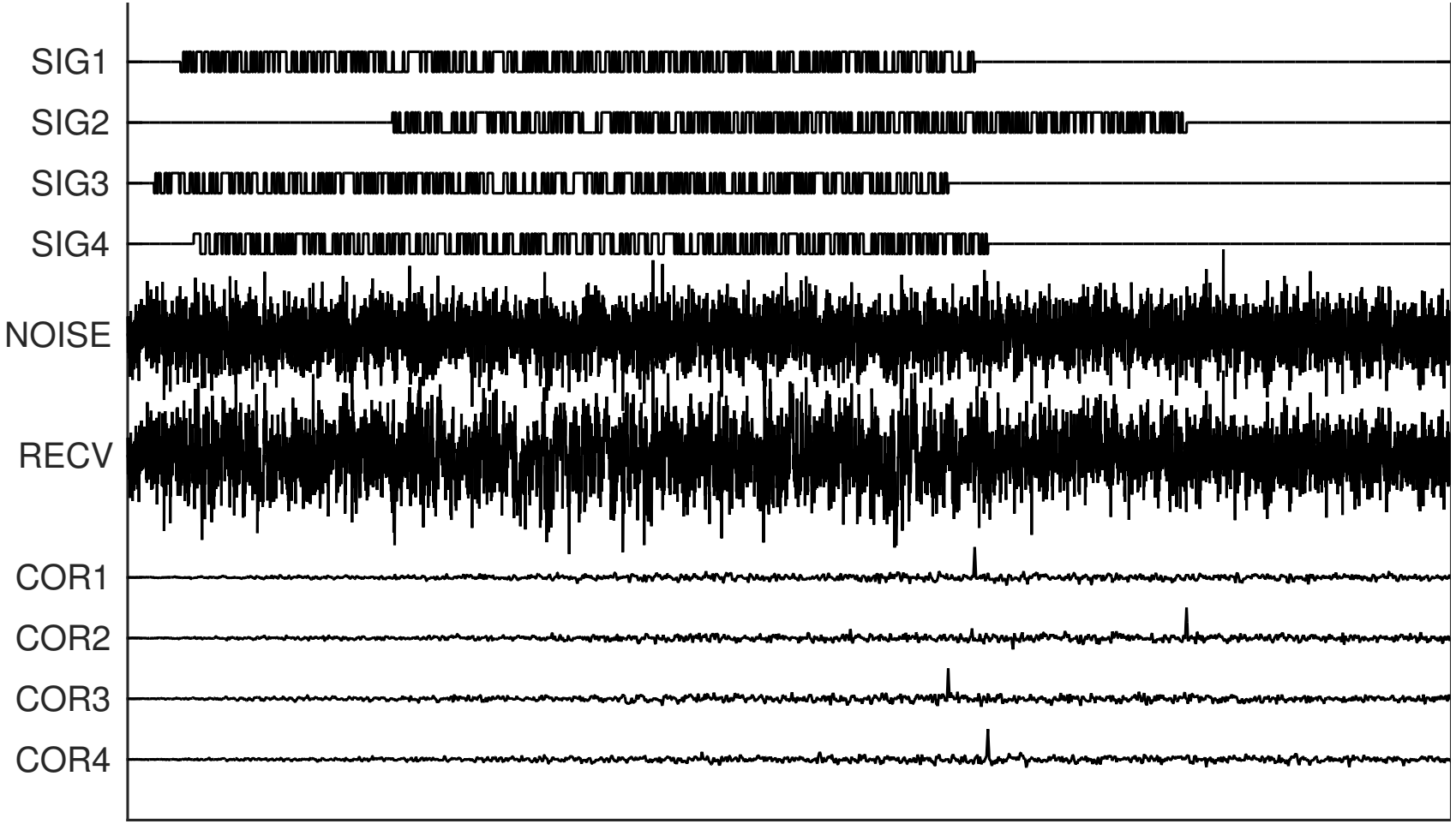
- Asymmetric security for satellite navigation signals
- Two-way distance-bounding protocols
- Tamper-resistant hardware

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



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

Existing technology:

- GPS/Galileo open access channel:
  highly predictable signal $\Rightarrow$ everyone can fake the satnav signal
- GPS military channel: encrypted spreading sequence
  military receivers know private key $\Rightarrow$ insider can still 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 codes / Digital signatures

Message authentication code (MAC)

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

Typical MAC lengths: 24...96 bit

Digital signature

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

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

Hash chains

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, ..., R_{n-1} with $R_{i-1} = h(R_i)$ or equivalently

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

- 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 $M_1, M_2, \ldots, M_n$:

- t_0 : $\text{Sign}_{PK}(t_1, t_i - t_{i-1}, R_0), R_0$ 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 : $(\text{Mac}_{R_3}(M_2), M_2, R_2)$ where $R_2 = h(R_3)$
- t_3 : $(\text{Mac}_{R_4}(M_3), M_3, R_3)$ where $R_3 = h(R_4)$
- t_4 : $(\text{Mac}_{R_5}(M_4), M_4, R_4)$ where $R_4 = h(R_5)$
- ...

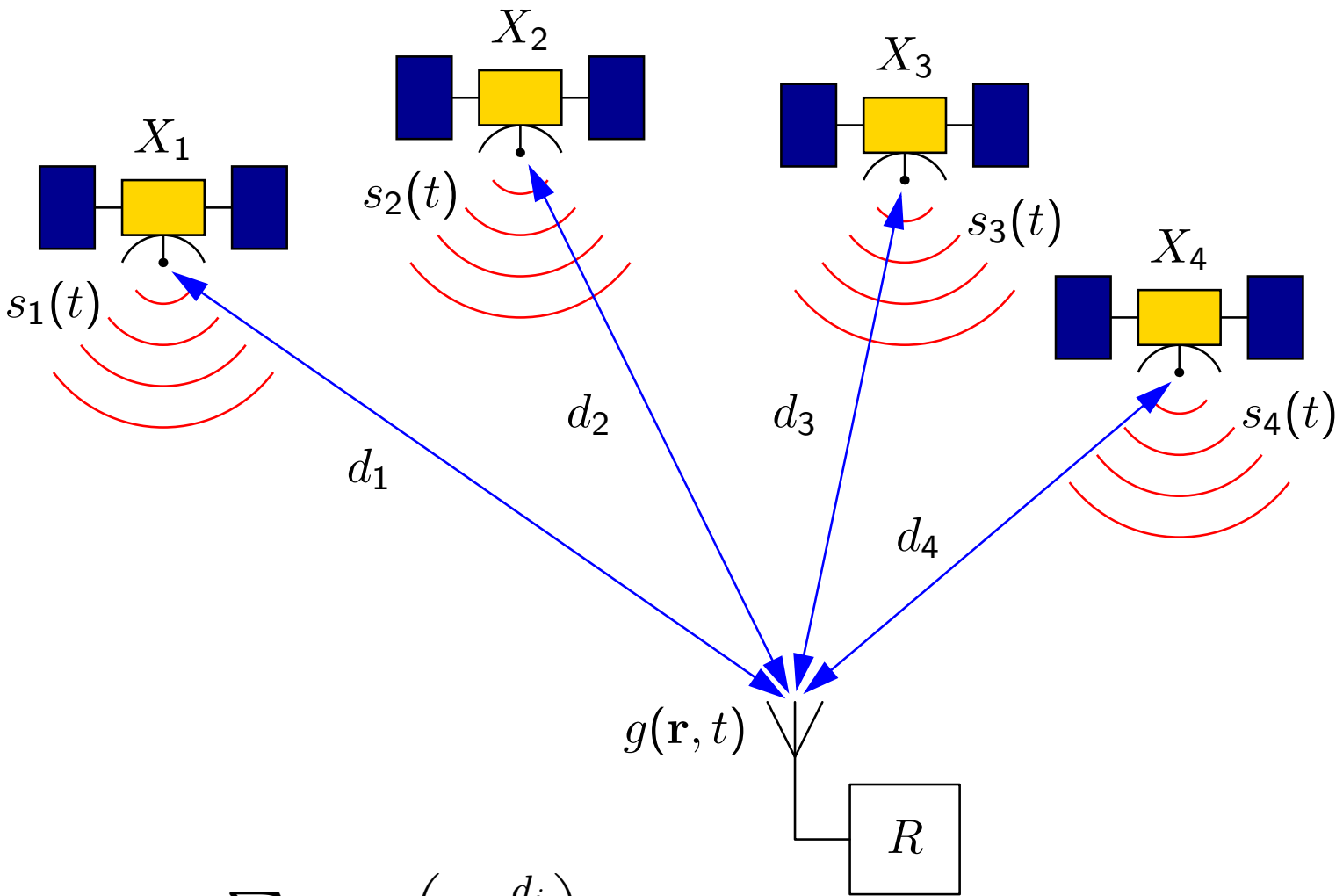
Each R_i is revealed at a pre-agreed time t_i. The MAC for M_i can only be verified after t_{i+1} when key R_{i+1} is revealed.

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

All MAC bits and initial bits of R_i are unpredictable. Note that the final bits of R_i could be brute forced at low transmission rates, and therefore cannot be considered unpredictable.

Using $R_i = h(t_i, R_{i+1})$ instead eliminates the risk of hash chain entering a cycle.

Pseudorange positioning systems



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

Attacks on trusted navigation receivers I

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.

Attacks on trusted navigation receivers II

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.

Prerequisite:

Attacker needs to decompose $$g(\mathbf{r}, t) = \sum_i A_i \cdot s_i \left(t - \frac{d_i}{c} \right)$$

Countermeasures:

- Give receiver a high-accuracy local trusted clock precise to detect delay Δt.
- make it difficult to decompose signal into contributions from different satellites

Attacks on trusted navigation receivers III

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

Navigation signal = Spreading code $\times$ navigation message

GPS C/A signal:

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

Equivalent distances:

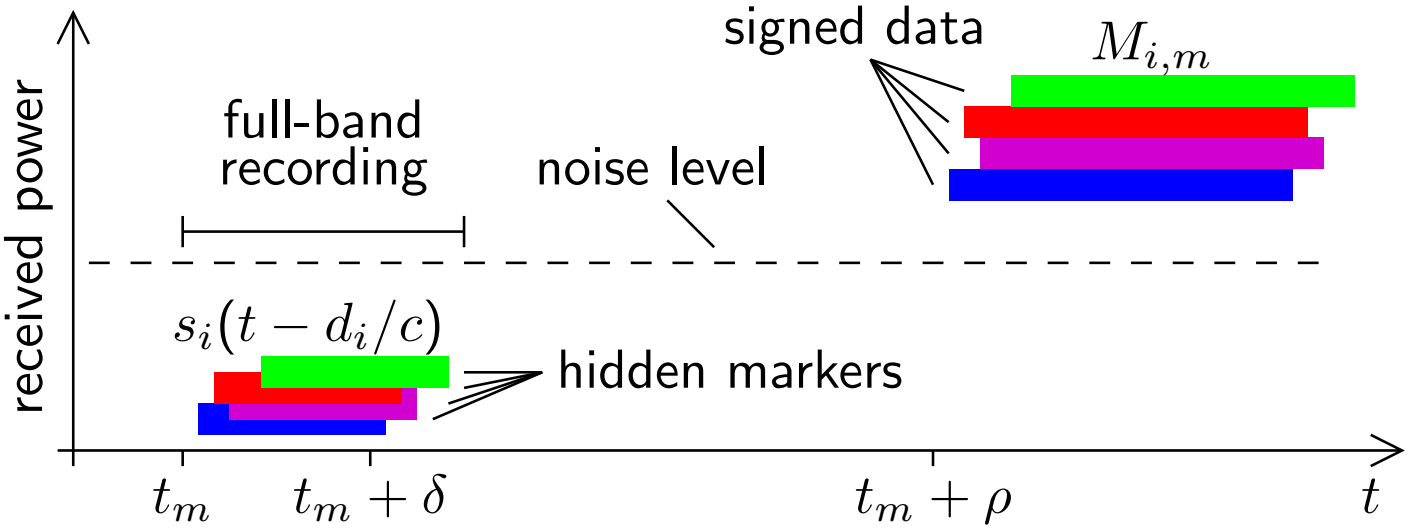
- 1 chip = 1 μs = 300 m
- 1 PRN cycle = 1 ms = 300 km
- 1 bit = 20 ms = 6000 km

Galileo OS similar: $\approx$2 Mbit/s, 200 bit/s

Asymmetric satnav security through hidden markers:

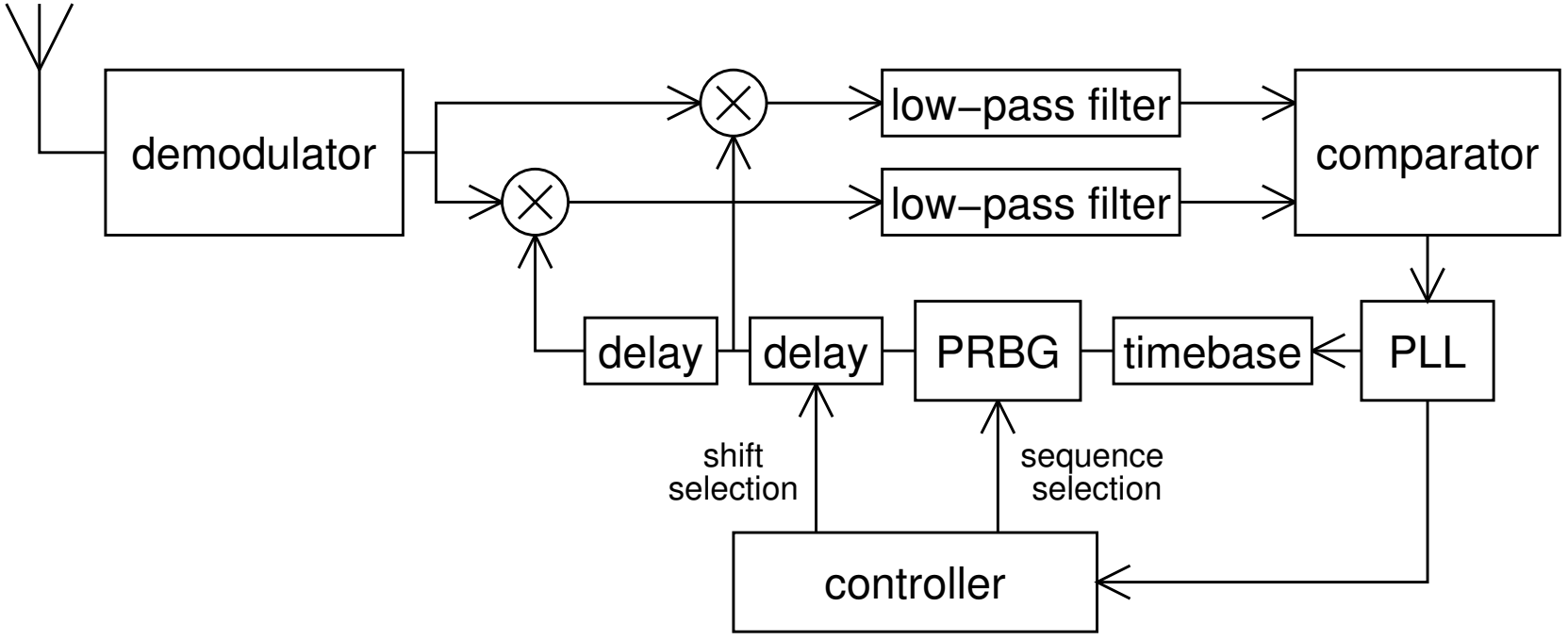
This solution is a steganographic process:

- 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



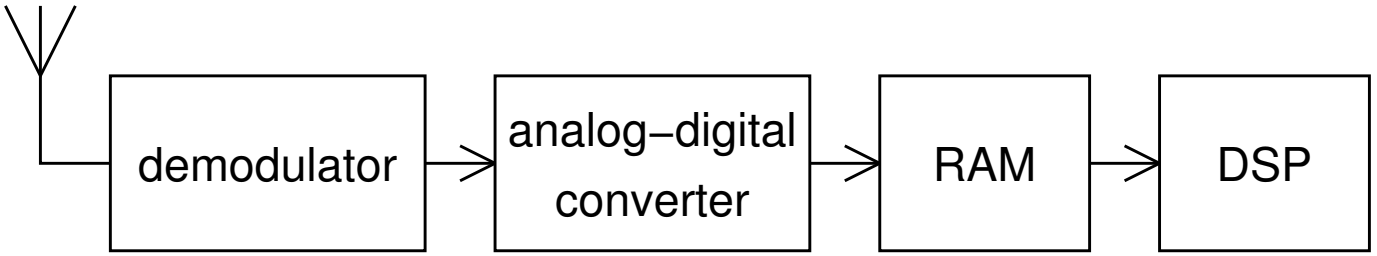
GPS receiver

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 $\Rightarrow$ 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).

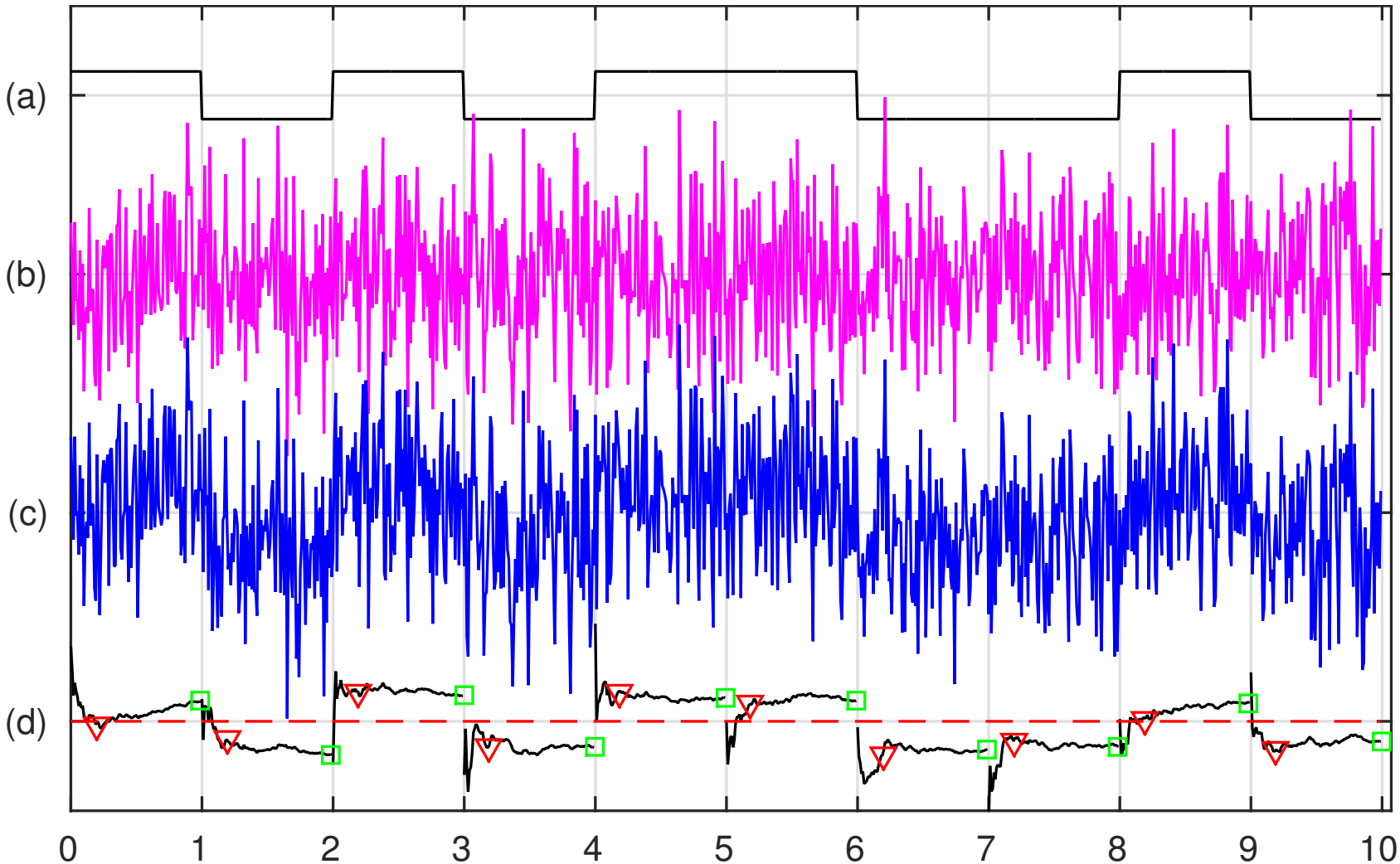
Switching to an initially secret spreading code is a major change to a GNSS system, not easily achieved in the near future.

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

Problems:

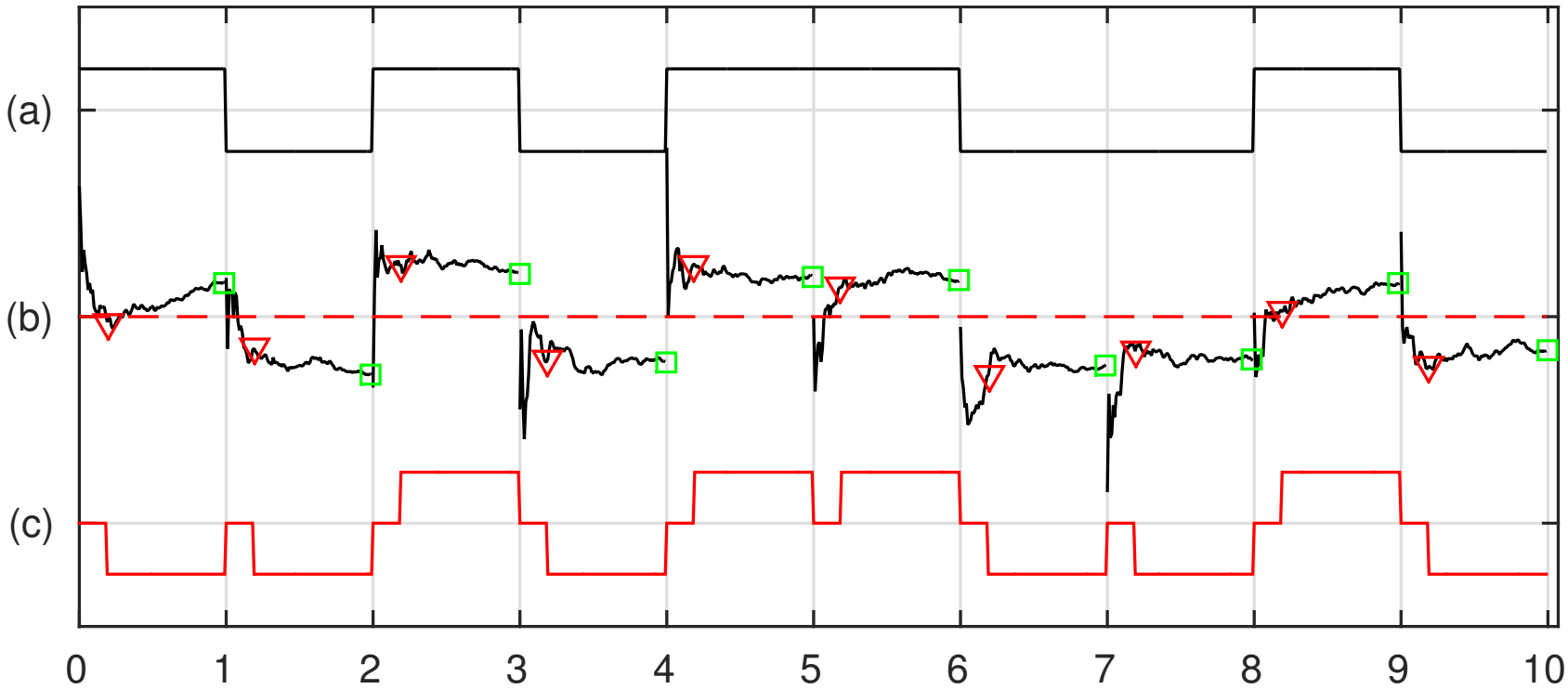
- unpredictable bits appear at a significantly lower rate (10^6)
- an attacker can perform early detection
  - a normal receiver detects data bits using a matched filter that integrates the signal over the full duration of the data bit $\rightarrow$ lowest bit error rate
  - a spoofer can operate a receiver that detects data bits after integrating only an initial fraction of the duration of the data bit $\rightarrow$ 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



(a) transmitted signal, (b) channel noise, (c) received signal, (d) integrator output in detector

Special detector provides early bit estimate



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

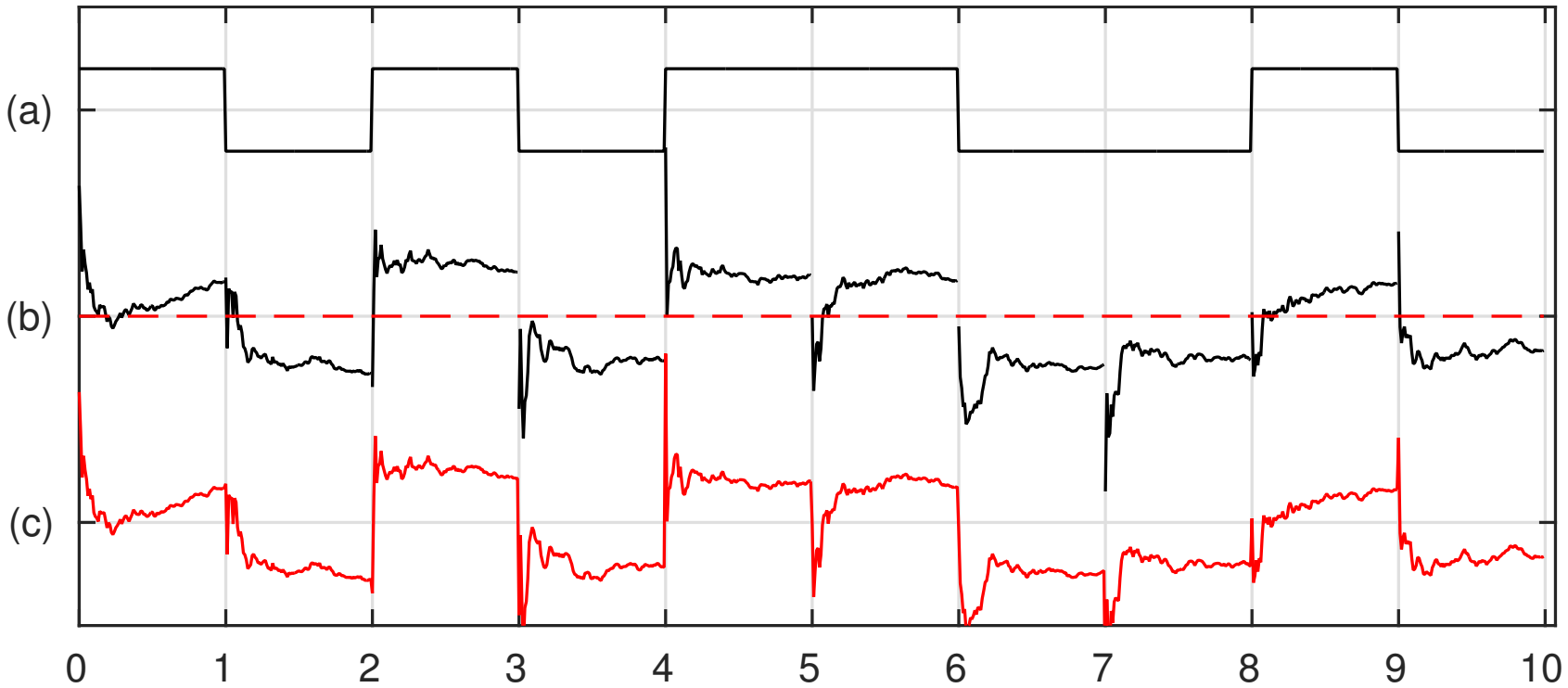
Early decision:

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

- 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

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

Special detector provides early bit estimate



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

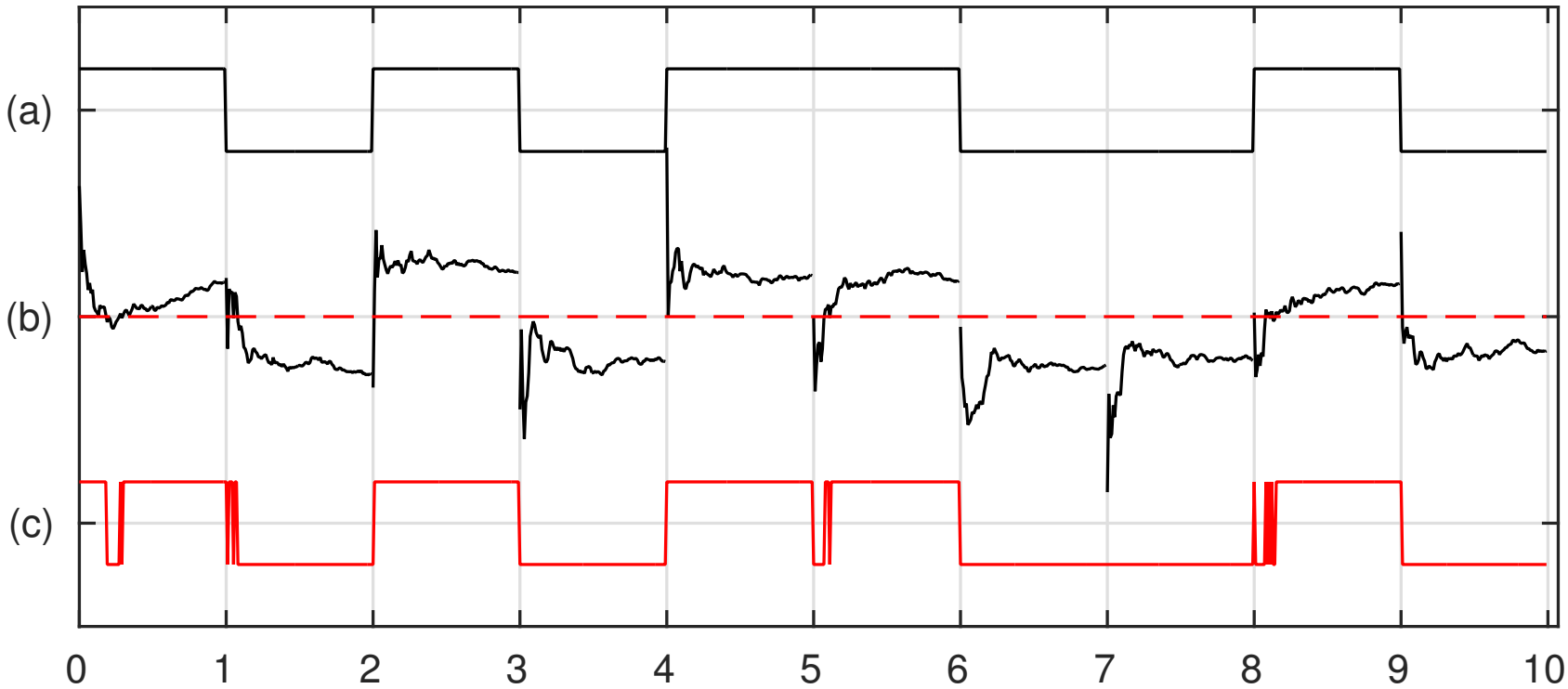
Maximum likelihood estimate (ML): $b(t) = (t - t_0)^{-1} \cdot \int_{t_0}^{t} a(t)\, dt$

- Continuously output current integrator value:

$$b(t)$$

[Humphreys 2013]

Special detector provides early bit estimate



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

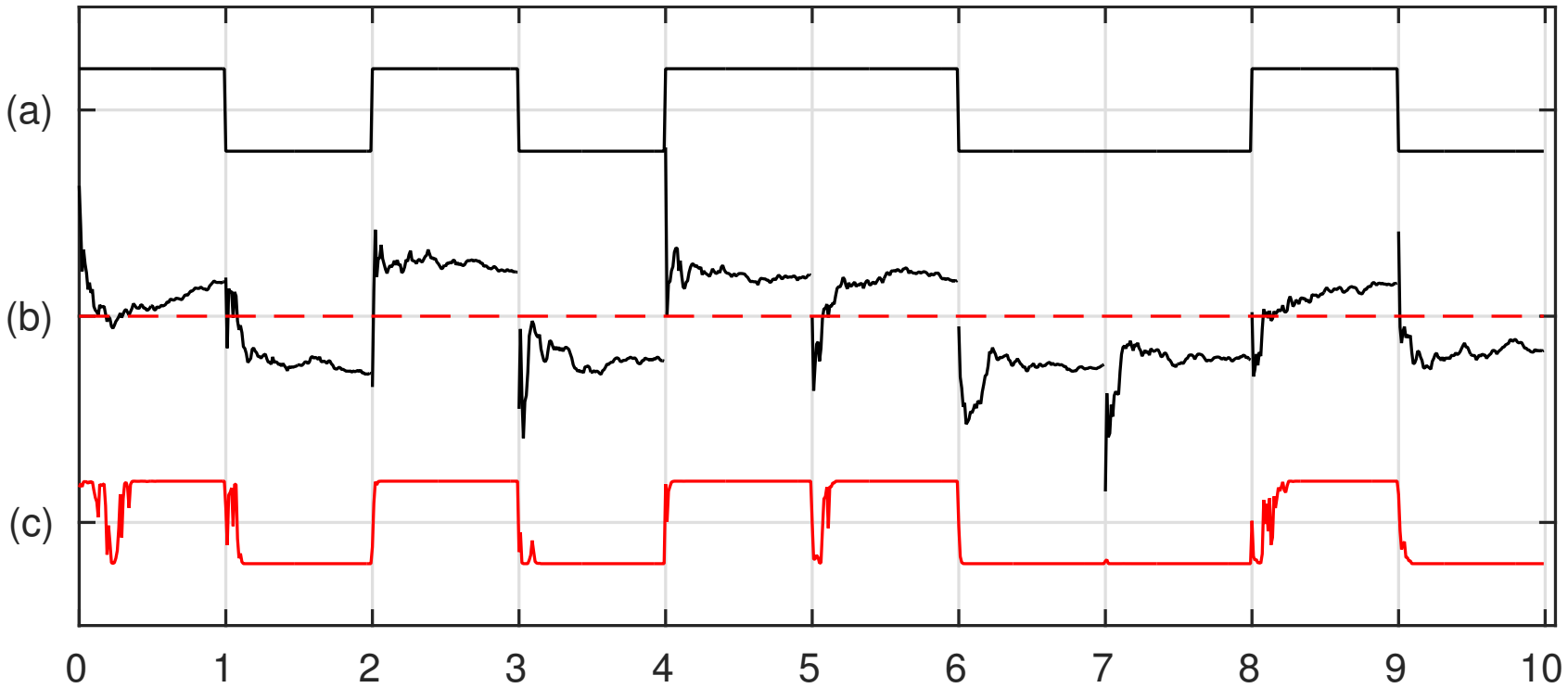
Maximum aposteriori probability (MAP): $b(t) = (t - t_0)^{-1} \cdot \int_{t_0}^{t} a(t)\, dt$

- Continuously apply threshold to current integrator value:

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

[Humphreys 2013]

Special detector provides early bit estimate



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

Minimum mean-square error (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: don't expect too much

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

Conclusions I

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 µs) 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 µs) 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 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.

`https://www.cl.cam.ac.uk/research/security/`