Protecting GNSS-based Services using Time Offset Validation

Kewei Zhang
Networked Systems Security Group
KTH Royal Institute of Technology
Stockholm, Sweden
kewei@kth.se

Marco Spanghero
Networked Systems Security Group
KTH Royal Institute of Technology
Stockholm, Sweden
marcosp@kth.se

Panagiotis Papadimitratos
Networked Systems Security Group
KTH Royal Institute of Technology
Stockholm, Sweden
papadim@kth.se

***Abstract*—Global navigation satellite systems (GNSS) provide pervasive accurate positioning and timing services for a large gamut of applications, from Time based One-Time Passwords (TOPT), to power grid and cellular systems. However, there can be security concerns for the applications due to the vulnerability of GNSS. It is important to observe that GNSS receivers are components of platforms, in principle having rich connectivity to different network infrastructures. Of particular interest is the access to a variety of timing sources, as those can be used to validate GNSS-provided location and time. Therefore, we consider off-the-shelf platforms and how to detect if the GNSS receiver is attacked or not, by cross-checking the GNSS time and time from other available sources. First, we survey different technologies to analyze their availability, accuracy and trustworthiness for time synchronization. Then, we propose a validation approach for absolute and relative time. Moreover, we design a framework and experimental setup for the evaluation of the results. Attacks can be detected based on WiFi supplied time when the adversary shifts the GNSS provided time, more than $23.942\,\mu s$; with Network Time Protocol (NTP) supplied time when the adversary-induced shift is more than $2.046\,ms$. Consequently, the proposal significantly limits the capability of an adversary to manipulate the victim GNSS receiver.**

***Index Terms*—Time Cross-checking, WiFi, NTP, Replay, Spoofing**

I. Introduction

The recent increased use of global satellite navigation systems (GNSS), for emerging applications, such as autonomous/unmanned vehicles or intelligent transportation systems has heightened security concerns. More so, as researchers recently demonstrated an effective GPS spoofer built with a Raspberrry Pi and a Software-Defined Radio (SDR), with a cost of only $250 [1]; or a dual-frequency spoofer built with an SDR with a cost of only $400 [2]. Therefore, any applications relying on GNSS, from mainstream mobile devices to smart vehicles, ships and large, complex systems, such as smart grids and cellular networks, face a dire risk.

A significant effort to improve security against different types of attackers, such as GNSS repeaters and spoofers, has been a central focus for both industry and the research community. GNSS vulnerabilities have been investigated in several works, e.g., [1] and [3]–[6], and different countermeasures have been analyzed and evaluated [7]–[16]. Contributions towards protecting GNSS receivers can be divided into two main categories: countermeasures on the receiver side and on the system side. On the receiver side, one approach to detect the presence of an attacker is to check the received signal strength, e.g., through received power monitoring (RPM) ([8], [17]) and automatic gain control (AGC) monitoring [18]; with special purpose hardware a receiver can determine the arriving angles of the signals from different satellites [19]–[21]; some work compares GNSS measurements with additional positioning information, e.g., Inertial Navigation System (INS), to detect the spoofing or replaying attacks [22]–[24]; distortions of signal correlation function [25]–[27] and clock drift ([10], [17]) can also be an indication of an attacker.

On the system side, modification of the GNSS infrastructure is needed, to add or augment features of Signal in Space (SIS), to increase the difficulty of mounting attacks. Military signals can be encrypted with secret keys that can be accessed only by authorized entities ([7], [28]). For civilian-grade signals, Galileo currently develops navigation message authentication (NMA) for Open Service (OS) Signals [29]–[31]; other systems use similar approaches to protect civilian signal authenticity [7]. However, even with NMA protection, signals can be still manipulated by sophisticated replay attacks, such as distance-decreasing attacks ([4], [32], [33]) and secure code estimate-replay attacks [30], which can modify each pseudorange measurement separately.

It is feasible to detect the attack by checking the consistency of the GNSS position, velocity, and time (PVT) solution. The aforementioned INS-based countermeasures [22]–[24], necessitate INSs. Without such hardware, it is not feasible for many applications to detect abnormalities in the PVT solution obtained from the GNSS receiver. Methods to detect GNSS spoofing attacks by checking the clock bias, within a short period were proposed ([10], [17], [34]). The detection method is based on a known linear clock state model with stable clock drift. However, it was found that the receiver's clock drift becomes stable only after about 120 minutes after switching on the receiver in room temperature, because it takes about 100 minutes for the receiver temperature to become stable [17]. This approach is not easily applicable, less so for a receiver in cold start.

Currently, many commercial devices/platforms with an embedded GNSS receiver have rich connectivity, by means of different technologies, notably WLAN and cellular networks.

This leads to another path for time/clock information to be used as means to detect attacks. The approach is to cross-check timing information with external time sources. Therefore, we can leverage these technologies to obtain several different external time sources, to detect if the GNSS-provided time is consistent with them. Based on this, if the external timing information source is not attacked. i.e., if it can be trusted, it is possible to determine whether the received GNSS signals are legitimate or not. The effectiveness of this approach depends on the time accuracy provided by different technologies, as discussed in Section II. Although the idea to use time as a mean of verification is not new, we propose, to the best of our knowledges, a first investigation towards generalizing the comparison of different time sources to detect discrepancies, between time provided by the GNSS and external (non-GNSS) technologies. Our experimental setup, based on commercially available off-the-shelf (COTS) devices, is a general test setup to evaluate the performance with real data.

The rest of the paper is organized as follows: Sec. II analyzes the time accuracy of different technologies; then, the adversary model is presented in Sec. III, following by our proposed algorithm in Sec. IV; furthermore, a test setup and evaluation results are in Sec. V; finally, Sec. VI concludes the work.

II. Related Technologies

A GNSS receiver can be just one component in a system/platform that offers many network connectivity options. These different connections can be leveraged to obtain different external time sources independent from each other.

The Network Time Protocol (NTP) and the Precision Time Protocol (PTP) provide accurate clock synchronization over LAN/WAN networks and they are the industry standards for synchronization in computer systems ([35], [36]). Recently, several security concerns, especially man-in-the-middle attacks and denial of service attacks, were investigated [37]–[40]. NTPsec is a security-hardened implementation of NTP, which aims to make the protocol deployment compliant with more stringent security, availability, and assurance requirements [41]. The accuracy of NTP is usually within tens of milliseconds over the Internet, and it can be less than 1 millisecond in LANs with ideal network conditions. However, asymmetric network conditions and routes degrade NTP accruacy to 100 milliseconds or more ([42], [43]). PTP suffers from similar problems, but it provides better accuracy, from hundreds of nanoseconds to microseconds [44]. In contrast to their good performance over wired links, using these protocols over mobile communication links raises a series of challenges. One of the known problems in implementing NTP over cellular networks is the change of state of the cellular radio. If the amount of traffic on the communication link is not enough to keep the radio in active state, the radio goes in idle mode. As specified in the 3GPP documentation [45], when the cellular radio is forced into a idle mode, no physical uplink or downlink are allocated. The power state transition introduces significant communication latency and it degrades the performance, limiting its accuracy. Keep-Alive messages are needed to generate enough traffic for the connection to avoid idle states [46]. The achievable accuracy is within tens of ms ([46], [47]) with strong constraints on the power consumption and operational modes of the cellular radio. In scenarios where power consumption is a critical limitation, this solution is hardly applicable.

Cellular networks are widely deployed, including 2G, 3G, 4G and now 5G, providing comprehensive network access coverage in cities, highways and countryside. From the development of 3G and 4G to 5G, highly accurate time synchronization has become available. Timing Advance (TA) values, used to schedule transmissions between User End (UE) and Radio Base Station (RBS), are used to synchronize UE and RBS since 2G ([48], [49]). The TA value is normally between 0 and 63, with iincrements of 3.69 microseconds, i.e., one bit period. This value also defines the best accuracy the UE can obtain through the TA values. For LTE, Release 11 of the LTE standard defines a new System Information Block (SIB), i.e., SIB16, which contains GPS time and Coordinated Universal Time (UTC), so that the UE uses them to obtain GPS and UTC time or local time [50]. In 5G, two proposals for UE time synchronization methods in RAN#81 leverage a SIB-based message, i.e., SIB16, to deliver reference time information to UEs for Time Sensitive Networking (TSN) ([51], [52]). The worst case synchronization inaccuracy with a Next Generation Node B (gNB) is expected to be ± 250 ns for small Industrial Internet of Things (IIoT) cells (e.g., up to 10 m radius) [52].

Cellular links are not the only option to access high precision timing signals. WiFi and other Wireless LAN-based technologies offer several solutions. In [53], experiments show that average propagation delay using NTP over WLAN is 2.7 ms with a standard deviation of 2.39 ms. Some customized WLAN protocols ([54], [55]) propose storing timestamps inside Beacons, so that there is no protocol overhead in establishing synchronization between Access Point (AP) and mobile stations. It is also proposed to store many older timestamps in beacons, to ensure reliability in case of beacon loss. The accuracy these synchronization algorithms achieve is around 100 μs, with a customized driver on a Windows platform. For wireless distributed systems, synchronization of internal clocks is a fundamental problem to allow communication. Protocols such as Reference Broadcast Time Synchronization (RBS), perform well in distributed scenarios [56]. In urban environments, the high density of Access Points (APs) can provide seamless WiFi beacon coverage. This large number of beacons, generated by APs to advertise their networks, can be exploited to provide a stable flow of high-precision timing information. However, as the probability of receiving several streams of beacons is significant (given the dense deployment of APs, e.g., in urban environments), the computational power needed to process all such events can become a significant bottleneck for a low end platform. To avoid this, a subset of the available beacons, based on the proximity to the AP and the target beacon emission rate can be selected.

Local clock references in state-of-the-art CPUs [57] can be used to provide a very stable time, based on the timestamp instruction cycle register of the CPU. Specifically, the Time Stamp Counter (TSC) (and equivalent), a 64-bit processor register, counts the number of CPU clock cycles since reset. Therefore, the TSC value maintains very high time resolution, e.g., one nanosecond for a stable $1\,\mathrm{GHz}$ processor. When the TSC is used for accurate timing, the speed/frequency of the CPU needs to be controlled and kept stable. Intel allows developers to extract TSC information since the Pentium CPU [58]; similarly for AMD processors [59] and ARM processors [60]. Performance Measurement Units (PMUs) or clock registers can be read from the Linux kernel and the user space from ARM, AMD and Intel processors. Even though platform specific, these registers are common to several architectures of the same family.

III. System Model

A. Adversary Model

The mathematical model to obtain the PVT solution at a GNSS receiver influenced by an adversary can be written as:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{f} + \mathbf{v} \tag{1}$$

where $\mathbf{y}$, $n \times 1$ vector, contains pseudorange measurements of the receiver to n satellites; $\mathbf{H}$ is a $n \times 4$ observation matrix; $\mathbf{x} = [x, y, z, \delta t]$, is the receiver state, including three-dimension coordinates and clock offset; $\mathbf{f}$ is a $n \times 1$ offset vector that an adversary introduces to the pseudorange measurements; $\mathbf{v}$, $n \times 1$ vector, is noise. Generally speaking, there is no limitation on how the adversary mounts the attacks, e.g., by replaying previously recorded signals or by transmitting fine-grained simulated signals, and the adversary objectives.

Considering a simple adversary, all the elements in $\mathbf{x}$, including $x,\ y,\ z$ and δt, are manipulated when the adversary induces a specific victim receiver location, or the adversary could seek to mislead the receiver to follow the adversary-intended time, by inducing a specific δt. Then, our and any time cross-checking proposal will detect the attack when the change in δt exceeds a certain threshold. The objective is to design time-based validation (or attack detection) that operates in a way that severely limits the adversary. Intuitively, the approach detects the lowest discrepancy caused by an attack. A sophisticated adversary, seeking to change the victim's location without modifying the victim's time [61] cannot be detected by any clock-related countermeasure, and thus by our proposal either.

The GNSS receiver may be at cold start or in a state of continuously tracking satellites. In the cold start, the system needs to acquire absolute time from satellites or other external time providers. When the GNSS receiver already tracks satellites, the adversary can either first jam the signals reception at the receiver, then transmit recorded/forged signals to the receiver, or use a signal lift-off technique to take over the receiver.

B. System Assumptions

Maintaining synchronization of the GNSS receiver as accurate as possible is not the goal of our proposal. Instead, we aim at evaluating to which extent the existing external time transfer technologies can be used towards the GNSS-provided time and location verification. When the external time technologies are used to detect the manipulation of the GNSS-provided information, clear assumptions on the trustworthiness and accuracy metrics are needed:

- It is not likely that the adversary can attack the victim GNSS receiver and at the same time compromise the external time sources or manipulate the access to the external, non-GNSS, time sources, e.g., NTP servers. Therefore, for this work, we assume that external time sources are trusted. For example, the network access per se can be encrypted and authenticated, or the time-providing network server or component (e.g., access point or base station) can be authenticated.
  Remark: In the event of no trusted external non-GNSS time sources, the approach would amount to a discrepancy detection between GNSS-provided time and one or more external time sources. Intuitively, such a discrepancy detection between two essentially non-trusted sources of time can still be useful: it can reveal that either the GNSS or the external time source(s) is attacked. This more complex attack surface warrants its own investigation; we discuss this briefly in Sec. VI.
- The GNSS receiver is always the primary synchronization source, unless it is deemed not trusted by the validation/detection scheme.
- The system always chooses the most accurate time source. The only exception: another available source more trusted than the default one, even if the latter is less accurate.
- The system will always synchronize with the most trusted available source, informing the user about any changes in time reference.

IV. Solution Approach

Without loss of generality, we consider two situations: 1) there is only one external time technology, e.g., due to limited connectivity or functionality; 2) multiple available external time technologies are available. Moreover, the approach can be developed in two different directions: 1) validating relative time that is, the difference of the GNSS and external technologies elapsed time during the same time interval; 2) validating absolute time, that is, the difference of absolute time from each technology (GNSS or not). For the sake of simple presentation, we discuss first the case of a single available external technology. Then, we extend it to the case of multiple available external technologies.

A. Single Available External Technology

In general, the device/system can access different external time sources/technologies, e.g., WLAN or cellular networks, each providing a different level of accuracy. However, due

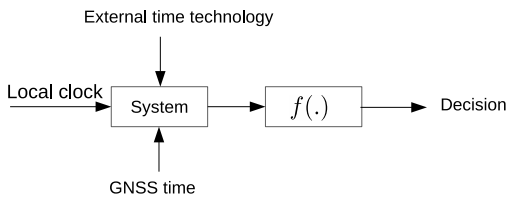


Fig. 1: Illustration of the approach with a single external time source

to environment limitations and other constraints, there might be only one available external source. As Fig. 1 shows, the system applies a function, $f(.)$, to the GNSS time and the one provided by the external technology. The output indicates whether the GNSS time is consistent with the external time sources.

This function, $f(.)$, can be implemented based on two approaches:

- Validating absolute time, T
- Validating relative time, $\Delta t = T_n - T_{n-1}$, where n is the index of GNSS time update

1) Absolute time checking:

$$f(t) = f(|T_{ext}(t) - T_{GNSS}(t)|) \tag{2}$$

is a function of the time difference between GNSS and the external technology. Specifically, T_{ext} is the time value from the external source, T_{GNSS} is the GNSS-provided time, and t is a time instance at the system when both T_{ext} and T_{GNSS} are available.

The receiver starts acquiring time information from the GNSS, in cold or warm start, and simultaneously acquires time from the external technology; it updates its GNSS-provided time every τ seconds, with the value, τ, depending on the design of the receiver, e.g., $500\,\mathrm{ms}$ or $1\,\mathrm{s}$. For each GNSS update, there is a time fetch from the external time technology. We do not consider the accuracy of GNSS-provided time, i.e., around $100\,\mathrm{ns}$, which means that the GNSS-provided time is deemed accurate in the absence of an adversary. Therefore, we have:

$$f(t) < \epsilon_{ext} \tag{3}$$

where ϵ_{ext} is the external time technology accuracy, subject to the network delays, the wireless propagation environment or the external time source attached master clock.

For each available technology, there can be multiple time information sources. For instance, our enhanced receiver can acquire NTP time from several different NTP servers; or it can receive WiFi beacons from multiple access points simultaneously. Therefore, assuming there are k time sources for a single technology, we have series of $f(t)$ values for each time source:

$$\begin{matrix} f^1(t_1) & f^1(t_2) & \dots & f^1(t_n) \\ & \vdots & & \\ f^k(t_1) & f^k(t_2) & \dots & f^k(t_n) \end{matrix} \tag{4}$$

By assuming the k sources of one technology (having same attributes and thus expected time accuracy), we set a counter m incremented at each time instance t_n, if Eq. 3 is true:

$$\text{For } i = 1, ..., k; \qquad m = m + 1 \quad \text{if } f^i(t_n) < \epsilon_{ext} \tag{5}$$

Then, the approach makes a decision at time t_n based on:

$$\frac{m}{k} > \frac{1}{2} \text{ (or a desired higher value)} \tag{6}$$

which indicates that the majority of sources of the available time technology satisfy Eq. 3, i.e., the technology-specific accuracy threshold. If Eq. 6 is not true, this indicates a discrepancy between time acquired by GNSS and the external technology.

2) Relative time checking:

$$f(t) = f(|\Delta t_{ext}(t) - \Delta t_{GNSS}(t)|) \tag{7}$$

where $\Delta t_{ext}(t) = T_{ext}(t+1) - T_{ext}(t)$ and $\Delta t_{GNSS}(t) = T_{GNSS}(t+1) - T_{GNSS}(t)$. The idea of relative-time checking is that, given one interval measured by GNSS-provided time, the elapsed time measured by the external technology should be within a certain threshold. In absence of an adversary, $f(t)$ satisfies the following:

$$f(t) < \epsilon_{ext} \tag{8}$$

The validation process is similar as described in Eqs. 4 and 6.

For both absolute-time and relative-time checking, in order to reduce the false alarm probability, we can extend this scheme to an aggregated scheme: the approach makes one decision every Q time instances. When any Q successive events give negative results based on Eq. 6, an attack or discrepancy of GNSS-provided time and the external technology provided time is signaled.

B. Multiple External Available Technologies

Multiple external technologies can be available in many off-the-shelf platforms. As Fig. 2 shows, during the system bootstrapping phase, the system searches and locks to available satellites, thus obtains PVT solutions. Meanwhile, it acquires time information from other external sources. The system has a predefined setting about the accuracy (and trustworthiness, in the next version of this work, as currently external time sources are deemed trusted) of different external time technologies, according to historical statistics.

Therefore, for absolute-time and relative-time checking at time instance t, $g(t)$ is defined as follows:

$$\begin{aligned} g(t) &= g\{|T_{ext1}(t) - T_{GNSS}(t)|, \dots, |T_{extk}(t) - T_{GNSS}(t)|\} \\ g(t) &= g\{|\Delta t_{ext1}(t) - \Delta t_{GNSS}(t)|, \dots, \\ &\qquad |\Delta t_{extk}(t) - \Delta t_{GNSS}(t)|\} \end{aligned} \tag{9}$$

When we trust all the external non-GNSS time sources (of different types/technologies), if the majority of those time technologies fulfills Eq. 6, the approach deems that GNSS provided time is not faulty.

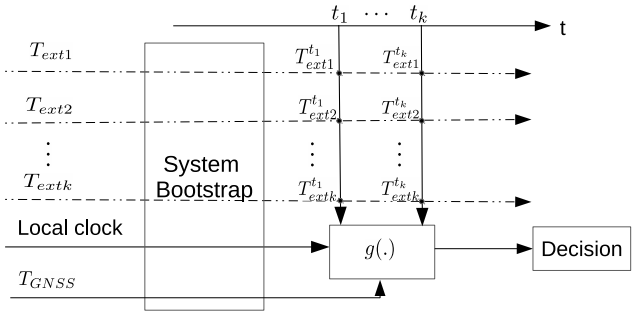


Fig. 2: Illustration of the approach with multiple external time sources

If we do not fully trust the external time technologies, the approach applies a weight to each time technology, by considering their level of trustworthiness and accuracy. For instance, in an industrial environment, the trusted WLAN access points have higher weight than cellular networks. Hence, the decision at each time instance is made based on:

$$g(t) = w_{ext1}\frac{f_{ext1}(t)}{\epsilon_{ext1}} + w_{ext2}\frac{f_{ext2}(t)}{\epsilon_{ext2}} + \cdots + w_{extk}\frac{f_{extk}(t)}{\epsilon_{extk}} < 1 \tag{10}$$

where w_{extk} is the weight of kth time technology and $\sum_{i=1}^{k} w_{exti} = 1$ for k external time technologies, and $f(t)$ can be a function based on absolute time, Eq. 2, or relative time, Eq. 7. The weights, w_{extk}, can be defined based on the trustworthiness and accuracy of external time sources and the conservativeness of threshold ϵ_{extk}. In order to reduce the false alarm probability, aggregation of results of successive Q decisions, to obtain one final decision on the attack, can be used.

V. Architecture and Evaluation

A. Framework/Architecture

To evaluate the concept and demonstrate the results, we designed an architecture compatible with multiple external time technologies, as presented in Fig. 3a. We have a centralized controller that interacts with the system clock and all other time technologies, including GPS, WiFi beacons, NTP servers, etc. The system detection is triggered by the GPS time update within a specified interval; the system makes a detection decision at each specified interval with the available collected data.

One of the challenges is the combination of synchronous and asynchronous data collection processes from different time sources. The reason is that when the system triggers a detection event, both for absolute time verification and relative time verification, all the collected data must be aligned, in order to compare them with each other. More specifically, data collection of WiFi beacons is asynchronous due to their spontaneous transmission characteristics; NTP data collection is synchronous because the NTP request is on-demand when the system attempts a request.

The solution is to apply a time alignment for different time technologies, especially for the asynchronous data collection. We use the platform/system clock as a reference to align the data; the system timestamps each data collection with its local clock and compensates for delays of data provided by different time technologies.

B. Experimental Setup

We leverage two external time services, NTP and WiFi beacons, to verify the GPS time, as presented in Fig. 3b. The ublox EVK-6T evaluation kit [62] offers two interfaces for data transmission: a USB2 port provides a real-time PVT solution that the manipulated GPS time is synthesized based on; a RS232 serial port provides a GPS time pulse for the host synchronization. The serial port provides a high accuracy Pulse Per Second (PPS) via the Data Carrier Detect (DCD) pin, which is used as a reference to check the performance of the evaluation results.

Beyond the GPS receiver, the rest of the configuration includes:

- Host machine: Intel I7 CPU running a Linux system whose kernel supports high precision timing.
- Host WiFi card: Intel Corporation Wireless-AC 9260.
- WiFi beacons: from surrounding access points of the office building.
- NTP servers: three servers in Sweden.
- GPS PVT rate: 1 Hz.
- Observation window candidates for APs: $T_{window} = \{1024,\ 3072\ ,5120\}$ ms.

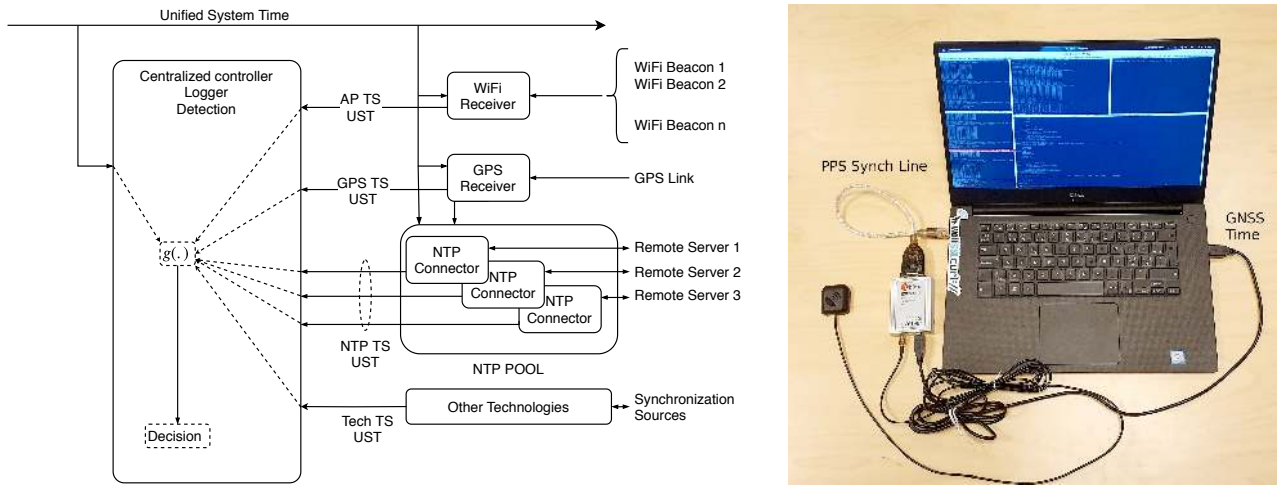
C. Evaluation

1) Accuracy Analysis: To validate the GPS time, we need to obtain the accuracy, ϵ_{ext}, of each technology, as shown in Eqs. 3 and 8. The accuracy is obtained by comparing the time information from each technology with a GNSS disciplined oscillator.

For the case of NTP, the accuracy is calculated based on the offset between the GNSS-provided time and the NTP server provided time. The left plot of Fig. 4 shows the offset of three different NTP servers located in the same country, for a period of 22 hours. We use the 99% quantile of each server offset as its accuracy, as shown in the right plot of Fig. 4, to represent our parameter ϵ_{ext}. The highest value among the three ϵ_{ext} is chosen to set the threshold for the NTP time. Therefore, we have:

$$\epsilon_{NTP} = 2.046 \text{ ms} \tag{11}$$

A similar approach is used to profile WiFi beacons. By default, APs transmit beacons at a 100 Time Unit (TU) interval which corresponds to 1024 microseconds [63]. In the IEEE 802.11 standard, a timestamp field contained in each beacon indicates the time, notably T_{AP}^{B}, the beacon leaves the AP. Specifically, T_{AP}^{B} is the elapsed time since



(a) Framework architecture

(b) Measurement and evaluation setup

Fig. 3: Framework architecture and demonstration setup

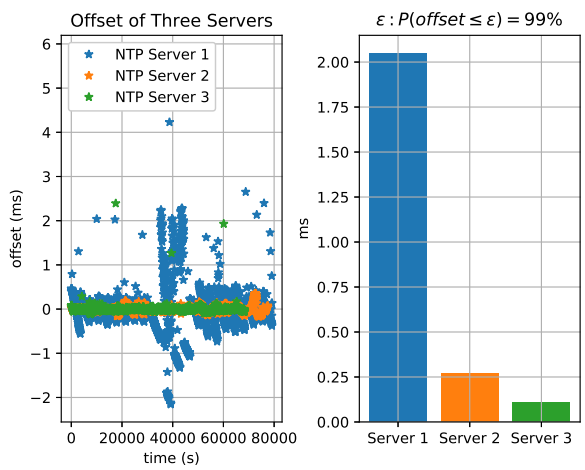


Fig. 4: Statistics of three NTP servers

power-up of the radio interface. The accuracy of the WiFi beacons is calculated over three different observation windows, $T_{window} = \{1000,\ 3000,\ 5000\}$ TU. First, the difference between the timestamp of the received beacon at the end of the observation window and the timestamp of the beacon at the beginning of the window is $T_{AP}^{B_{end}} - T_{AP}^{B_{begin}}$. Then, the error between this difference and the T_{window} interval measured by the GNSS-disciplined clock is used to determine the accuracy for WiFi beacons, as presented in Fig. 5. The left plot is the error of recorded AP data during each T_{window} compared to the one measured by the GNSS disciplined clock, and the right plot gives their 99% quantiles.

With a similar approach to the one used to evaluate the NTP accuracy, we choose the highest value of the quantiles among the three APs for each T_{window} as the accuracies:

$$\epsilon_{WiFi} = \begin{cases} 46.064\ \mu s & T_{window} = 1024 \text{ ms} \\ 35.021\ \mu s & T_{window} = 3072 \text{ ms} \\ 23.942\ \mu s & T_{window} = 5120 \text{ ms} \end{cases} \tag{12}$$

2) Evaluation Results: The synthesized GPS time we use is obtained by applying an offset function, $of(.)$, to the real GPS time at step n:

$$\begin{aligned} of(1) &= t_{bias} \\ of(2) &= of(1) + \beta \\ &\vdots \\ of(n-1) &= of(n-2) + (n-3) * \beta \\ of(n) &= of(n-1) + (n-2) * \beta \end{aligned} \tag{13}$$

where t_{bias} is the initial offset to the real GPS time, β controls the rate of increment of the offset and n indicates the nth GPS update. Fig. 6 shows our synthesized GPS attack, t_{bias} =5 µs and β =0.055 µs. After time $t_0 = 3492$, the offset is maintained constantly, i.e., 360 ms.

For the evaluation based on relative time checking with WiFi beacons, for the three different observation windows, the system selects the GPS time samples every $\{1,\ 3,\ 5\}$ updates, then it picks beacons within the window as follows:

$$\begin{aligned} &\text{start beacon } B_1\text{: } |T_{RX}^{B_1} - T_{window}^{start}| < 100 \text{ TU} \\ &\text{end beacon } B_2\text{: } |T_{RX}^{B_2} - T_{window}^{end}| < 100 \text{ TU} \end{aligned} \tag{14}$$

where T_{RX}^{B} is the system reception time at the receiver, and T_{window}^{start} and T_{window}^{end} are the system instants of GPS updates at the beginning and end of the window.

When beacons satisfying the above requirements cannot be found, the system triggers a detection at the next GPS time update. Otherwise, it applies $T_{AP}^{B_2} - T_{AP}^{B_1}$ as Δt_{ext} to Eq. 7 and tests the comparison against the accuracy threshold in Eq. 12. The test results are presented in Fig. 7; where we can see the scheme cannot detect the GPS attack in the beginning, before time t_1, when the offset is lower than ϵ_{WiFi} for $T_{window} =$

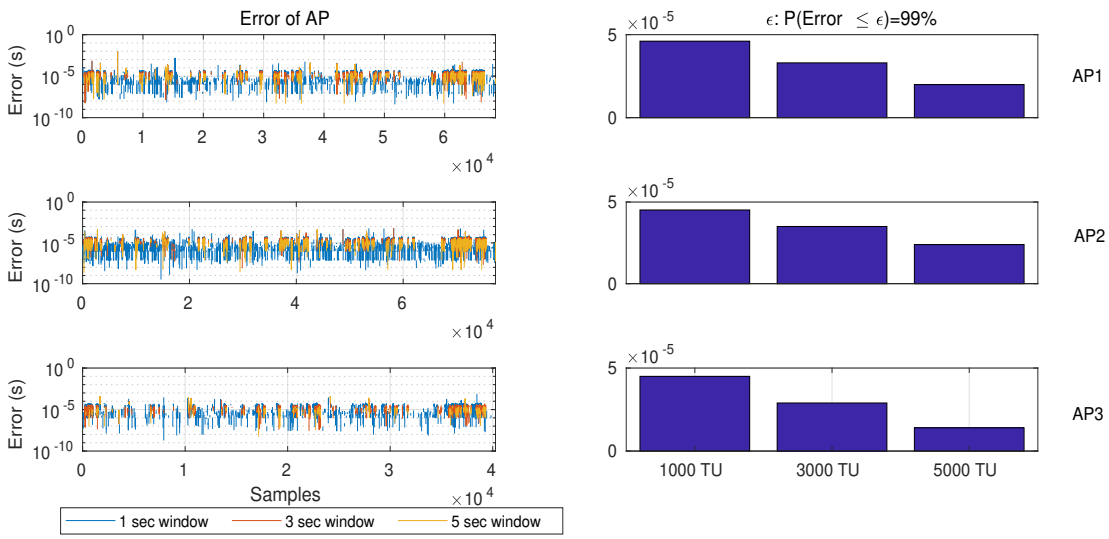


Fig. 5: Statistics of three access points (AP)

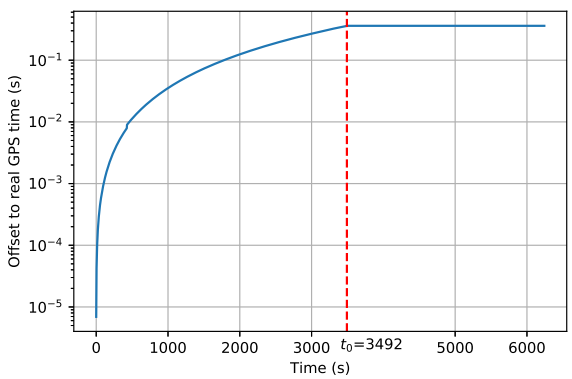


Fig. 6: The offset between real GPS time and synthesized GPS time

$\{1000,\ 3000\}$ TU. The scheme can detect the attack from time t_1 to t_0. But it can trigger an alarm of detecting the attack for $T_{window} = \{5000\}$ TU from the beginning of the attack to t_0. After time t_0, because the time offset is constant, the relative time checking solution with WiFi beacons is no longer effective.

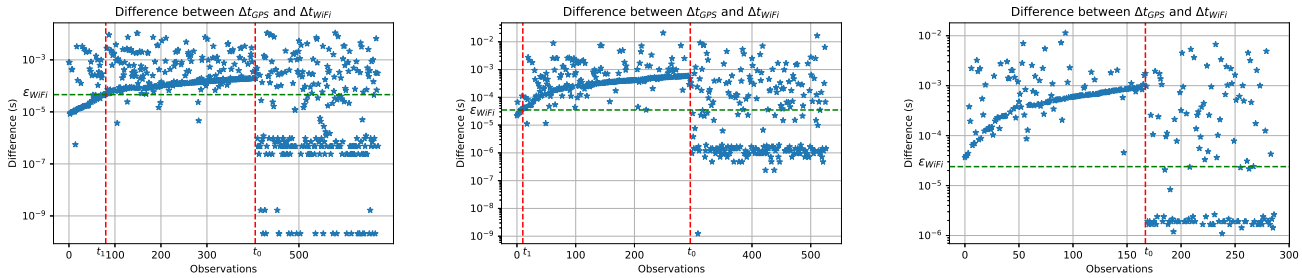
For the verification based on absolute time checking with NTP, the system acquires the NTP values at selected GPS samples, with a frequency lower than the minimum polling frequency specified by the NTP service provider. The system applies the acquired T_{ext} and corresponding T_{GPS} to Eq. 3, with the accuracy defined in Eq. 11. The comparison results for each NTP server are presented in Fig. 8: the system can detect the attack when the offset is higher than ϵ_{NTP}.

VI. Discussion and Future Work

We investigated to which extent other timing technologies can potentially protect a GNSS receiver from a replay/spoofing attack, based on their availability, diffusion and accuracy. We proposed a scheme and designed an experimental framework for the detection of these attacks. The scheme and the framework can integrate various timing technologies, providing a scalable and flexible solution based on time cross-checking. We demonstrated and evaluated the proposal with a real-world collected data, with different setup configurations, both for relative time checking and absolute time checking.

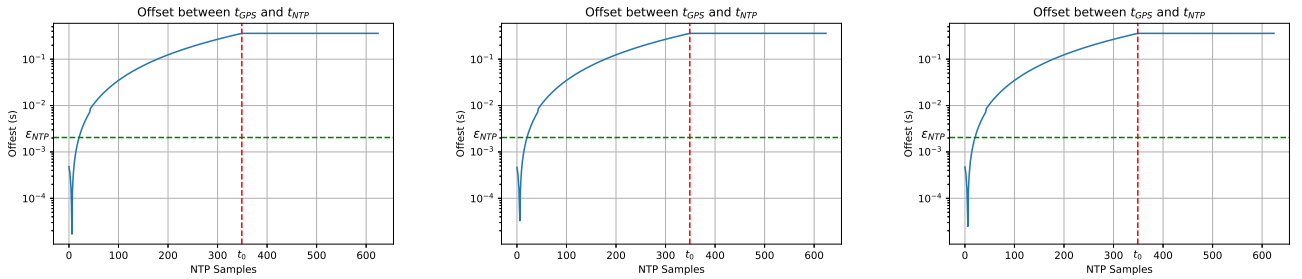
The proposed concept aims to protect a GNSS receiver using time information obtained from external sources and alternative independent technologies. One of the concerns that can arise is to which extent we can trust such external information. It is part of future investigation to determine how the level of trustworthiness of these technologies affects the security of the proposal. Without authentication of the WiFi beacons or authenticated network access or authenticated time servers, a substantial limitation would arise. Broadcasting bogus WiFi beacons is not hard, and it would be possible for an attacker to emulate a set of access points to transmit beacons with proper T^B_{AP}, which can mask the alteration in the manipulated GNSS time. Our framework can weigh alternative sources or even rank the verification external time sources, based not only on their precision but also on the perceived level of trust, as specified in Eq. 10 (Sec. IV-B).

The future version of the implemented system will consider the possibility of adopting more types of external time sources, along with varying levels of trustworthiness. As an example, Long Range (LoRA) [64] or IEEE 802.15.4 compliant networks [65], can provide low power consumption connectivity. These technologies are potential external time sources to enhance the verification capability of the system once the infrastructure is deployed and the technologies gets popular in mobile systems.



(a) $T_{window} = 1024\,\text{ms}$ (b) $T_{window} = 3072\,\text{ms}$ (c) $T_{window} = 5120\,\text{ms}$

Fig. 7: Verification results of WiFi beacons for different T_{window}



(a) Server 1 (b) Server 2 (c) Server 3

Fig. 8: Verification results for different NTP servers

Acknowledgments

Work supported by the Swedish Foundation for Strategic Research (SSF) SURPRISE project and the KAW Academy Fellowship Trustworthy IoT project.

References

[1] K. C. Zeng, S. Liu, Y. Shu, D. Wang, H. Li, Y. Dou, G. Wang, and Y. Yang, "All your GPS are belong to us: Towards stealthy manipulation of road navigation systems," in *27th USENIX Security Symposium*, 2018, pp. 1527–1544.

[2] J. T. Curran, A. Morrison, and C. ODriscoll, "(In)Feasibility of Multi-Frequency Spoofing ," https://insidegnss.com/infeasibility-of-multi-frequency-spoofing/, Accessed 25 Nov 2019.

[3] H. Kuusniemi, J. Blanch, Y.-H. Chen, S. Lo, and P. Enge, "Feasibility of Fault Exclusion Related to Advanced RAIM for GNSS Spoofing Detection," in *Proceedings of the ION GNSS*, 2017, pp. 2359–2370.

[4] K. Zhang and P. Papadimitratos, "On the Effects of Distance-decreasing Attacks on Cryptographically Protected GNSS Signals," in *Proceedings of the 2019 International Technical Meeting of The Institute of Navigation (ION ITM 2019)*, Reston, Virginia, 2019, pp. 363–372.

[5] G. Gibbons, "FCC fines operator of GPS jammer that affected Newark airport GBAS," *Inside GNSS*, vol. 30, 2013.

[6] K. Zhang and P. Papadimitratos, "Safeguarding nma enhanced galileo os signals from distance-decreasing attacks," in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, 2019, pp. 4041–4052.

[7] J. M. Anderson, K. L. Carroll, N. P. DeVilbiss, J. T. Gillis, J. C. Hinks, B. W. OHanlon, J. J. Rushanan, L. Scott, and R. A. Yazdi, "Chips-Message Robust Authentication (Chimera) for GPS Civilian Signals," in *ION GNSS*, 2017.

[8] D. M. Akos, "Who's afraid of the spoofer? GPS/GNSS spoofing detection via automatic gain control (AGC)," *Navigation*, vol. 59, no. 4, pp. 281–290, 2012.

[9] K. D. Wesson, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "An evaluation of the vestigial signal defense for civil gps anti-spoofing," in *Proceedings of the ION GNSS Meeting*, 2011.

[10] P. Papadimitratos and A. Jovanovic, "GNSS-based positioning: Attacks and Countermeasures," in *Military Communications Conference, 2008. MILCOM 2008. IEEE*. IEEE, 2008, pp. 1–7.

[11] ——, "Protection and Fundamental Vulnerability of GNSS," in *IEEE International Workshop on Satellite and Space Communications (IEEE IWSSC)*, Toulouse, France, October 2008, pp. 167–171.

[12] ——, "Method to Secure GNSS-based Locations in a Device having GNSS Receiver," April 2012, US Patent 8,159,391. [Online]. Available: http://www.google.com/patents/US8159391

[13] B. Motella, D. Margaría, and M. Paonni, "SNAP: An authentication concept for the Galileo open service," in *Proceedings of the 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Monterey, California, 2018, pp. 967–977.

[14] E. Shafiee, M. Mosavi, and M. Moazedi, "Detection of Spoofing Attack using Machine Learning based on Multi-Layer Neural Network in Single-Frequency GPS Receivers," *The Journal of Navigation*, vol. 71, no. 1, pp. 169–188, 2018.

[15] K. Zhang and P. Papadimitratos, "Secure Multi-constellation GNSS Receivers with Clustering-based Solution Separation Algorithm," in *2019 IEEE Aerospace Conference*. IEEE, 2019, pp. 1–9.

[16] Z. Gülgün, E. G. Larsson, and P. Papadimitratos, "Statistical Method for Spoofing Detection at Mobile GNSS Receivers," in *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2019, pp. 677–681.

[17] D. Marnach, S. Mauw, M. Martins, and C. Harpes, "Detecting Meaconing Attacks by Analysing the Clock Bias of GNSS Receivers," *Artificial Satellites*, vol. 48, no. 2, pp. 63–83, 2013.

[18] F. Bastide, D. Akos, C. Macabiau, and B. Roturier, "Automatic gain control (AGC) as an interference assessment tool," in *ION GPS/GNSS 2003, 16th International Technical Meeting of the Satellite Division of The Institute of Navigation*, 2003, pp. pp–2042.

[19] P. Y. Montgomery, T. E. Humphreys, and B. M. Ledvina, "Receiver-autonomous spoofing detection: Experimental results of a multi-antenna receiver defense against a portable civil GPS spoofer," in *Proceedings of the ION International Technical Meeting*, 2009, pp. 124–130.

[20] M. L. Psiaki, B. W. O'hanlon, S. P. Powell, J. A. Bhatti, K. D. Wesson, and T. E. Humphreys, "GNSS spoofing detection using two-antenna differential carrier phase," 2014.

[21] E. McMilin, Y.-H. Chen, D. S. De Lorenzo, S. Lo, D. Akos, and P. Enge, "Field test validation of single-element antenna with anti-jam and spoof detection," *Proc. ION GNSS+, Tampa, FL*, 2015.

[22] S. Khanafseh, N. Roshan, S. Langel, F.-C. Chan, M. Joerger, and B. Pervan, "GPS spoofing detection using RAIM with INS coupling," in *2014 IEEE/ION Position, Location and Navigation Symposium-PLANS 2014*. IEEE, 2014, pp. 1232–1239.

[23] J. T. Curran and A. Broumendan, "On the use of low-cost IMUs for GNSS spoofing detection in vehicular applications," in *Proceedings of the ITSNT*, 2017.

[24] C. Tanil, S. Khanafseh, and B. Pervan, "An INS monitor against GNSS spoofing attacks during GBAS and SBAS-assisted aircraft landing approaches," in *Proceedings of the 29th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2016)*, 2016, pp. 2981–2990.

[25] M. Pini, B. Motella, and M. T. Gamba, "Detection of correlation distortions through application of statistical methods," in *Proceedings of the 26th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2013)*, 2013, pp. 3279–3289.

[26] K. Ali, E. G. Manfredini, and F. Dovis, "Vestigial signal defense through signal quality monitoring techniques based on joint use of two metrics," in *2014 IEEE/ION Position, Location and Navigation Symposium-PLANS 2014*. IEEE, 2014, pp. 1240–1247.

[27] A. J. Jahromi, A. Broumandan, S. Daneshmand, G. Lachapelle, and R. T. Ioannides, "Galileo signal authenticity verification using signal quality monitoring methods," in *2016 International Conference on Localization and GNSS (ICL-GNSS)*. IEEE, 2016, pp. 1–8.

[28] GSA, "Galileo Commercial Service Implementing Decision enters into force ," https://www.gsa.europa.eu/newsroom/news/galileo-commercial-service-implementing-decision-enters-force, Accessed 25 Nov 2019.

[29] G. Caparra, S. Sturaro, N. Laurenti, C. Wullems, and R. T. Ioannides, "A Novel Navigation Message Authentication Scheme for GNSS Open Service," in *ION GNSS*, vol. 2016, 2016.

[30] 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.

[31] K. Wesson, M. Rothlisberger, and T. Humphreys, "Practical cryptographic civil GPS signal authentication," *Navigation*, vol. 59, no. 3, pp. 177–193, 2012.

[32] K. Zhang and P. Papadimitratos, "GNSS receiver tracking performance analysis under distance-decreasing attacks," in *Proceedings of the 2015 International Conference on Location and GNSS (ICL-GNSS 2015)*, Gothenburg, Sweden, 2015, pp. 1–6.

[33] K. Zhang and P. Papadimitratos, "Safeguarding NMA Enhanced Galileo OS Signals from Distance-Decreasing Attacks," in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, Miami, Florida, 2019, pp. 4041–4052.

[34] A. Jafarnia-Jahromi, S. Daneshmand, A. Broumandan, J. Nielsen, and G. Lachapelle, "PVT solution authentication based on monitoring the clock state for a moving GNSS receiver," in *European navigation conference (ENC)*, vol. 11, 2013.

[35] D. Mills, E. J. Martin, J. Burbank, and W. Kasch, "RFC5905 Network Time Protocol Version 4: Protocol and Algorithms Specification," in *Internet Engineering Task Force (IETF)*, 2010.

[36] I. S. Association *et al.*, "Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *IEEE 1588*, 2002.

[37] E. B. Haberman, D. Mills, and U. Delaware, "RFC5906 Network Time Protocol Version 4: Autokey Specification," in *Internet Engineering Task Force (IETF)*, 2010.

[38] M. Bishop, "A security analysis of the NTP protocol version 2," in *Proceedings of the Sixth Annual Computer Security Applications Conference*. IEEE, 1990, pp. 20–29.

[39] A. Liska, *NTP Security: A Quick-Start Guide*. Apress, 2016.

[40] T. Mizrahi, "Security Requirements of Time Protocols in Packet Switched Networks (2014)," 2017.

[41] E. S. Raymond, "Ntpsec: a secure, hardened NTP implementation," *Linux Journal*, vol. 2016, no. 270, p. 1, 2016.

[42] D. Mills, "RFC1305 Network Time Protocol Ver3 Specification, Implementation and Analysis," in *Internet Engineering Task Force (IETF)*, 1992.

[43] D. L. Mills, *Computer network time synchronization: the network time protocol on earth and in space*. CRC press, 2016.

[44] S. T. Watt, S. Achanta, H. Abubakari, E. Sagen, Z. Korkmaz, and H. Ahmed, "Understanding and applying precision time protocol," in *2015 Saudi Arabia Smart Grid (SASG)*. IEEE, 2015, pp. 1–7.

[45] Ericsson, "TSG-RAN Working Group 2 (Radio layer 2 and Radio layer 3)," Tech. Rep. August, Accessed 2020-02-01. [Online]. Available: https://www.3gpp.org/ftp/tsg_ran/WG2_RL2/TSGR2_06/docs/Pdfs/r2-99807.pdf

[46] H. Haverinen, J. Siren, and P. Eronen, "Energy consumption of always-on applications in WCDMA networks," in *2007 IEEE 65th Vehicular Technology Conference-VTC2007-Spring*. IEEE, 2007, pp. 964–968.

[47] R. Miskinis, D. Jokubauskis, D. Smirnov, E. Urba, B. Malysko, B. Dzindzeleta, and K. Svirskas, "Timing over a 4G (LTE) mobile network," 01 2014.

[48] 3GPP, "Radio subsystem synchronization TS 05.10," https://itectec.com/archive/3gpp-specification-ts-05-10/, Accessed 2020-02-01.

[49] ——, "GSM/EDGE Radio subsystem synchronization TS 45.010," https://itectec.com/archive/3gpp-specification-ts-45-010/, Accessed 2020-02-01.

[50] G. T. R2-125992, "Broadcast of Time Info by Using a New SIB v11.1.0," https://www.3gpp.org/DynaReport/36331-CRs.htm, Accessed 2020-02-01.

[51] G. T. R2-1817172, "Overview of UE Time Synchronization Methods," https://www.3gpp.org/DynaReport/TDocExMtg–R2-104–18808.htm, Accessed 2020-02-01.

[52] G. T. R2-1817173, "Clock Accuracy Realization at UE," https://www.3gpp.org/DynaReport/TDocExMtg–R2-104–18808.htm, Accessed 2020-02-01.

[53] D. Anand, D. Sharma, Y. Li-Baboud, and J. Moyne, "EDA performance and clock synchronization over a wireless network: Analysis, experimentation and application to semiconductor manufacturing," in *2009 International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*. IEEE, 2009, pp. 1–6.

[54] M. Mock, R. Frings, E. Nett, and S. Trikaliotis, "Continuous clock synchronization in wireless real-time applications," in *Proceedings 19th IEEE symposium on reliable distributed systems SRDS-2000*. IEEE, 2000, pp. 125–132.

[55] G. Cena, S. Scanzio, A. Valenzano, and C. Zunino, "A unified clock synchronization protocol for infrastructure wireless LANs," in *2015 IEEE 1st International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*. IEEE, 2015, pp. 508–515.

[56] A. Mahmood, R. Exel, H. Trsek, and T. Sauter, "Clock synchronization over IEEE 802.11 - A survey of methodologies and protocols," pp. 907–922, apr 2017.

[57] A. Pásztor and D. Veitch, "PC based precision timing without GPS," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 30, no. 1. ACM, 2002, pp. 1–10.

[58] Intel, "Using the RDTSC Instruction for Performance Monitoring," https://www.ccsl.carleton.ca/ jamuir/rdtscpm1.pdf, Accessed 2020-02-01.

[59] AMD, "BIOS and Kernel Developer's Guide for AMD Athlon^{TM} 64 and AMD Opteron^{TM} Processors," https://www.amd.com/system/files/TechDocs/26094.PDF, Accessed 2020-02-01.

[60] ARM, "Summary and description of the DWT registers," https://developer.arm.com/docs/ddi0337/e/system-debug/dwt/summary-and-description-of-the-dwt-registers, Accessed 2020-02-01.

[61] T. E. Humphreys, J. A. Bhatti, D. Shepard, and K. Wesson, "The Texas spoofing test battery: Toward a standard for evaluating GPS signal authentication techniques," in *Radionavigation Laboratory Conference Proceedings*, 2012.

[62] ublox, "EVK-6 u-blox Evaluation Kits User Guide," https://www.u-blox.com/sites/default/files/products/documents/EVK-6_UserGuide_(28GPS.G6-EK-10040).pdf, Accessed 2020-02-01.

[63] J. Geier, "802.11 Beacons Revealed," *Wi-Fi Planet*, 2002.

[64] Lora Alliance, "LoRaWAN 1.1 Specification," Accessed 2020-02-01. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf

[65] IEEE, "802.15.4-2015 - IEEE Standard for Low-Rate Wireless Networks," https://standards.ieee.org/standard/802_15_4-2015.html, Accessed 2020-02-01.