RAIM and SBAS based Detection of GNSS Spoofing by Timing and Content Consistency Rules

Guoyu Fu, Tyler Holmes, Colton Riedel, and Jyh-Charn Liu
Texas A&M University

BIOGRAPHIES

Guoyu Fu is a Ph.D. student in the Department of Computer Science and Engineering at Texas A&M University. He received his bachelor’s degree in Systems and Control from Nanjing University, China, in 2013. His research includes GNSS system modeling for high confidence requirements such as integrity checking, multipath avoidance, and computational geometry.

Tyler Holmes is a Ph.D. student in the Department of Computer Science and Engineering at Texas A&M University. He received his bachelor’s degree in Computer Science from Texas A&M University in 2017. His research interests include cybersecurity, reverse engineering, and software defined radios.

Colton Riedel is a Ph.D. student in the Department of Computer Science and Engineering at Texas A&M University. He received bachelor’s degrees in Mathematics and Computer Science from Texas Lutheran University in 2014. His research interests include high-performance computing, software defined radios, graph processing, and parallel and distributed computing.

Jyh-Charn Liu is a Professor in the Department of Computer Science and Engineering at Texas A&M University, College Station, Texas, USA. He earned his PhD from the University of Michigan in 1989, and his MS and BS from National Cheng Kung University, Tainan, Taiwan. His research interests span from system modeling and algorithms to parallel and distributed computer architectures.

ABSTRACT

In this paper, we analyze the GNSS spoofing problem from the perspectives of timing and content of GNSS messages. Among various types of GNSS attacks, we focus on the most powerful form of stealthy GNSS spoofing, based on the sequence of Receive, Modification, and Transmission (RMT) of GNSS messages. When precisely implemented, a RMT based spoofing attack can alter both the timing and content of GNSS messages at will, which can be very difficult for an unprepared regular receiver to detect and reject the bogus messages carried on the L1 channel. To address this open problem, in this work we explore high level software techniques to detect GNSS spoofing based on consistency checking among the timing and content of GNSS messages. We show that one can tailor the RAIM (Receiver Autonomic Integrity Monitoring) model to detect timing inconsistency provided that at least one authentic GNSS channel is intact. Next, we analyze the SBAS message broadcast system and estimate the challenge in SBAS spoofing. It is clear that it is significantly harder to spoof the timing alone of SBAS channels due to the dynamic content broadcast. Simulation and experiments on the TEBAT spoofing dataset were conducted to validate the proposed spoofing detection method.

1 INTRODUCTION

GNSS (Global Navigation Satellite Systems) signal spoofing refers to emission of RF signals on GNSS frequencies to imitate legitimate signals from space vehicles, with the goal of fooling victim receivers into accepting spoofed signals rather than legitimate ones. Various spoofing techniques have been demonstrated in recent years, and they can be loosely classified as GNSS signal simulators, meaconing transceivers, or receive-modify-transmit (RMT) techniques [1]. The first two classes of spoofing techniques are either not stealthy enough to defeat GNSS lock control algorithms, or their other signal signatures, e.g., surging of power level, may give away the presence of the spoofing code sequences in the L1 channel. Detection methods for these techniques have been demonstrated, but detection of RMT based spoofing remains an open problem as of the writing of...
demonstrates effectiveness of our method by simulation and using the TEXBAT systems is discussed including receiver architecture, message broadcast timing, pseudoranges, and RAIM. Analysis of SBAS reviews spoofing attacks and defenses in greater detail. In section 3 the importance and role of timing consistency in GNSS the performance of the

The first condition is deterministic in its nature, not based on dedicated high electronics available to the spoofer. The remainder of the paper is organized as follows: Section 2 reviews spoofing attacks and defenses in greater detail. In section 3 the importance and role of timing consistency in GNSS systems is discussed including receiver architecture, message broadcast timing, pseudoranges, and RAIM. Analysis of SBAS message type sequence and content for both WAAS and EGNOS SBAS satellites is presented in section 4. Lastly, section 5 demonstrates effectiveness of our method by simulation and using the TEXBAT spoofing datasets [4].
2 REVIEW OF RMT SPOOFING AND DEFENSE

The RMT spoofing attack is powerful because it can earn the victim receiver’s trust by fooling its Delay Lock Loop (DLL) to track the spoofed signal. In [1], the first version of a RMT spoofer manages to generate a spoofed signal with the same code phase as the authentic one. We use Figure 5 illustrate this process. In the first phase - “Shift-In”, this RMT spoofer finds the authentic code phase by searching in the range of a PRN code (e.g. 1 ms of GPS L1 C/A code). This searching process is a right-to-left shifting process, because the spoofed signal will be regarded as a multipath signal that comes later than the line-of-sight signal. It continues until the second phase - “Merge”, where the spoofed phase is code phase-aligned with the normal signal. The spoofed signal starts to take control of the DLL, as it overpowers the authentic signal. From this point, the spoofed signal arrives earlier than the normal signal in the “Pull-Off” phase, so that the normal signal will be regarded as multipath effects by the victim receiver. The RMT spoofer can control the length of pseudoranges for multiple channels of signals by shifting the spoofed code phase or modifying the message content. The spoofer can then mislead the victim to any desired position by changing the relative magnitudes of pseudoranges in multiple channels.

With regard to this spoofing process, many techniques are proposed to detect the existence of spoofing signals. Detection techniques based on signal characteristics include AGC (Automatic Gain Controller) based methods [2], angle of arrival based methods [3], message encryption based methods [5][6][7][8][9], Doppler and code phase relationships monitoring[10], vestigial signal defense methods [11], P(Y) code cross-correlation based methods [12][13], etc. Another category of detection techniques are based on analysis of the time and position relationship, including TDOA (Time Difference of Arrivals) based crowd-sourced method [13], cross-check with additional sensors [24], analysis of time series and spatial correlations [16][17] etc. These methods usually require extra hardware, or modification or access to low-level receiver hardware, which is not practical to be deployed in billions of existing GNSS receivers.

RAIM methods [18] and its extensions [19][20][21][22] have been considered in anti-spoofing works [23][24], but the RAIM model alone is not considered as a major scheme for spoofing detection. For example, in [25], the RAIM model is applied in a GPS/INS (Inertial Navigation System) integrated system, where the inertial sensor’s measurement is used as reference in the RAIM model to reveal GPS pseudorange anomaly. In addition to extra sensors, other GNSS constellations can be also used for reference measurements. For example, the SEL-2488 GNSS clock unit [32] detects spoofing by comparing the time information from GPS with GLONASS. Even though it is difficult for the spoofer to orchestrate multi-GNSS signals which are consistent with the real physics, it is not impossible as civilian signals from other GNSS (GLONASS, Galileo, Beidou, etc.) constellations have similar signal characteristics and predictable navigation messages.

3 GNSS SIGNAL TIMING CONSISTENCY MODELS

Figure 5: Three phases of the RMT spoofing

Note: these are make-up outputs of the DLL with many correlator taps
This section explains the normal timing relationship in a receiver (victim), and how an abnormal timing will mislead it to a wrong solution (position and time). We first review the general operation flow of a GNSS receiver, using Figure 2. At the antenna, a combination of signals from all GNSS satellite in view meet there, with a fixed timing relationship corresponding to the physical position and time of the antenna. At the frontend module, these UHF (Ultra-High Frequency) signals are down-converted to the IF (Intermediate Frequency) signal, and then chopped and quantized into digital samples by an ADC (Analog-Digital Converter). These samples, as still a mixture of all channels’ information, is replicated into each baseband channel (associated with each satellite signal) for separation. Tens of channels can be processed simultaneously by a modern GNSS receiver. In the acquisition module, each channel will first roughly search for the PRN (PseudoRandom Number) code phase and carrier frequency of the corresponding satellite. Then in the tracking module (code and carrier tracking), precise code phase and frequency are estimated, which are used to generate code and IF carrier replica for cancellation. The output of the tracking module is a bit stream of messages for the corresponding satellite.

Our following discussions are organized into two levels: the (pseudo) range-level and the solution-level. They both summarize timing information from the tracking loop, and data bits, but the range-level focuses on a single channel and the solution-level on all channels. The last part of this section describes the range based RAIM model which formally describes the relationship between these two levels of timing information.

### 3.1 Pseudorange-Level Timing Analysis

A pseudorange is not a natural measurement, but rather a computed result of two timing marks, the receiver measured arrival time \( t_{Ar} \), as well as the satellite scheduled transmit time \( t_{TS} \). Formally, the pseudorange measurement is expressed as

\[
\rho = C \cdot (t_{Ar} - t_{TS}), \quad (1)
\]

where \( C \) represents the speed of light in vacuum. The arrival time \( t_{Ar} \) is the reading of receiver’s clock when processing the current chunk of PRN codes in the DLL (Delay Lock Loop) code tracking module. The transmit time \( t_{TS} \) is a satellite time recorded in the signal. To calculate it, the receiver needs to find out the offset of the beginning of the current chunk of PRN codes with respect to the beginning of the current navigation message subframe, because the subframe starts at a known satellite time as given in the TOW (Time of Week) word.

We use Figure 3 to relate timing components in a pseudorange. In normal case, a pseudorange will contain normal errors: \( \delta_{sc} \)-the satellite clock bias (w.r.t. the universal GNSS time) at the physical transmission time \( t_{TS} \), \( \delta_{rc} \)-the receiver clock bias and noise at the physical arrival (at the antenna) time \( t_{A} \), \( \delta_{lo}^{l} \) and \( \delta_{tr}^{l} \)-the timing delays equivalent to the loss of speed during the travel in the ionosphere and troposphere. With these errors, a normal pseudorange is composed of:

\[
\rho^{l} = \sqrt{(x^{l} - x)^{2} + (y^{l} - y)^{2} + (z^{l} - z)^{2}} + C \cdot \delta_{sc}^{l} + C \cdot \delta_{lo}^{l} + C \cdot \delta_{tr}^{l} - C \cdot \delta_{sc}. \quad (2)
\]

In the abnormal case, we let \( j \) denote the polluted satellite signal. The pseudorange magnitude can be either smaller or larger than the normal value, since signal transmission time
can be later or earlier than the normal. Since the pseudorange is usually calculated based on the method of common-reception
time, Figure 3 only shows abnormal transmission time. But in reality, the spoofer may change both transmission time and/or
arrival time to achieve shortening or elongation of the victim’s range measurement. Specifically, the spoofer can do this via
two methods. First, the spoofed signal wavefront can come at the normal time, but its transmit time can be incorrect, since the
navigation message (ephemeris or satellite clock coefficients) can be modified by the spoofer. Second, the spoofed signal
wavefront comes at an abnormal time, either in advance of or behind the normal arrival time. In the case, the spoofer does not
need to change the data bits of the authentic signal, but only control the time of transmission from its antenna. Changing
message bits can be easily detected, but changing the time of arrival is more difficult to detect.

To change the time of arrival, the spoofer must earn the trust
of the DLL by sending signals that are of the same code
phase as the authentic signals. To be stealthy, the spoofing
signals at this stage should be identical to the authentic
signals, in terms of code phase, carrier phase, Doppler
frequency, and message contents. We use Figure 4 for
illustration, which is an artistic interpretation that imagines
the results for an infinite number of correlators. The DLL for
a channel \( j \), is supposed to have exactly one correlation peak
for the authentic PRN code. However, when being spoofed,
there is a ghost correlation peak for the spoofed code (the red box). It comes earlier than the authentic peak, so the DLL would
begin to track the spoofed code, and discard the authentic code as it is regarded a multipath-bounced code (as in the case of the
blue box). This spoofed case results in the red scenario in Figure 3, since the offset of the spoofed code to the start of navigation
subframe is smaller than the authentic code.

### 3.2 Navigation Solution-Level Model

The navigation solution module collects the satellite positions and pseudoranges of all channels into the system of navigation
equations as expressed in (3). In abnormal cases, the solution module mixes both the unpolluted and polluted satellites. We use
\( \mathbb{S} \) denote the set of all GNSS satellites in the field of view, \( \mathbb{S}_{\text{good}} \) and \( \mathbb{S}_{\text{bad}} \) the sets of unaffected authentic satellites and spoofed
satellites, respectively. In our discussion, we assume \( \mathbb{S}_{\text{good}} \neq \emptyset \). The system of navigation equations is as follows:

\[
\begin{aligned}
\bigcap_{i \in \mathbb{S}_{\text{good}}} \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 + c \cdot \delta_{rc}} = \rho_i, \\
\bigcap_{j \in \mathbb{S}_{\text{bad}}} \sqrt{(x - x_j)^2 + (y - y_j)^2 + (z - z_j)^2 + c \cdot \delta_{rc}} = \rho_j.
\end{aligned}
\tag{3}
\]

The geometric interpretation of (3) is illustrated in Figure 4. Let \( \mathbf{x} \) denote
an entity’s state (position and time), or \( \mathbf{x} = (x, y, z, c\delta_{rc})^T \). We draw three
normal satellite states \( (x^1, x^2, x^3) \), and their pseudoranges as the radius of the
black circles. They intersect at the physical location of the receiver’s
antenna, or the black dot \( \mathbf{x} \). Similarly, we draw three abnormal satellites in
red, assumed to be controlled by the spoofer. Their circles end up
intersecting another location, the red dot \( x' \), which is exactly the target
location that is intended by the spoofer. Given \( \mathbf{x}' \), the spoofer calculates the
corresponding relative magnitudes of pseudoranges, and schedule the
arrival times of the channels in control. In the extreme case where all
channels are under the spoofer’s control, all circles are red, and they
intersect at one fake location. Conversely, in the extreme case where all
channels are unpolluted, all circles are black and intersecting at the
receiver’s physical location. Our primary focus is cases between these two extremes exclusively.

In an abnormal situation where at least one but not all channels are affected by signal timing anomalies, the navigation system
solver plays as a role of “peacemaker” to compromise between normal and abnormal measurements. The commonly used
nonlinear system solver contains two parts: linearization and least square estimation. The linearization makes the position and
time calculation in an iterative fashion:
\[ \Delta \rho \approx H \Delta x, \quad (4) \]

where \( \Delta x = (\Delta x, \Delta y, \Delta z, C \Delta \delta)^T \) is the desired update amount to the position and time solution, \( \Delta \rho \) is an \( n \times 1 \) column vector containing \( \Delta \rho^1, \Delta \rho^2, \ldots, \Delta \rho^n \) for \( n \) channels’ pseudorange update, \( H \) is an \( n \times 4 \) matrix describing the directional geometry from the previous estimated receiver state to the current satellite positions. Then the receiver solves for the solution update \( \Delta x \), typically using the least-square solver. The least-square solver gets the likelihood-maximized estimation, denoted as \( \Delta x^* \), by minimizing the Euclidean norm of the difference vector for all channels:

\[ \Delta x^* = \arg \min_{\Delta x} \| \Delta \rho - H \Delta x \|^2 = (H^T H)^{-1} H^T \Delta \rho. \quad (5) \]

Let \( x^* \) denote the solved solution that is solved by accumulating previous estimations \( \Delta x^* \). As a result, \( x^* \) must be a point in the middle of the true state \( x \) and the fake state \( x' \), since the solved point can minimize the sum of squared residual measurement errors. As shown in Figure 5, the blue dot refers to the compromised solution by the least-square solver.

### 3.3 The Range-Based RAIM Model

The RAIM models formally describe the consistency in both the range-level and solution-level. Essentially, RAIM models reveal the discrepancy between the physical state \( x \) and the false state \( x' \) from range or solution perspectives [24]. They are commonly used to detect satellite failure, where only one or few satellites may cause timing anomaly. But in the spoofing scenario, the number of spoofed channels can be much greater than that of healthy channels. In this section, we reexamine the pseudorange residual based RAIM model in the presence of timing anomalies. Following the notations in section 2.2, we use \( x, x', \) and \( x^* \) to refer to true, spoofer planed and solved positions, respectively; \( s, i, \) and \( j \) the arbitrary, normal and abnormal channels, respectively.

Without loss of generality, we examine the Least-Squares-Residuals (LSR) method proposed by Lee [18], which is shown to be equivalent to two other range based RAIM methods [21]. The LSR method monitors the range residual of each channel, denoted as \( w \), and defined as the difference between a pseudorange and its estimate, denoted by \( \rho^* \) (a function of the solved solution \( x^* \)). The \( \rho^* \) in the LSR method is calculated based on the least square solution (see (4) and (5)). We use the following derivation process to show that the essence of the range residual is to scale the discrepancy between the solved solution \( x^* \) and the real solution \( x \) by a factor of the satellite-receiver’s cosine directions \( H \) (this process utilizes (4) and (5)):

\[ w = \rho - \rho^* = \rho - \rho_0 - (\rho^* - \rho_0) = \Delta \rho - \Delta \rho^* = H \Delta x - H \Delta x^* + \epsilon = H(x - x_0) - H(x' - x_0) + \epsilon = H(x - x^*) + \epsilon, \quad (6) \]

where the \( \rho_0 \) is some arbitrary pseudorange measurement at the state \( x_0 \). \( \epsilon \) is regarded as the measurement error vector, but it also contains second and higher-order linearization errors. Therefore, if \( x \approx x^* \) holds, the range residual \( w \) for each channel should be close to zero.

In a spoofing scenario, a healthy channel’s pseudorange is still associated with the receiver’s true state \( x \), but a spoofed channel’s pseudorange is with the spoofer-planned state \( x' \). The range residuals for a good channel \( i \) and a bad channel \( j \) will have different components:

\[ w^i = H^i(x - x^*) + \epsilon, \quad w^j = H^j(x' - x^*) + \epsilon. \quad (7) \]

This implies the range residual is proportional to the distance between the solved state and the two other states (true and spoofer-planned states). It also means when all channels are spoofed, there will be \( x^j \approx x^* \), and all range residuals are close to zero.
and redundant messages, thereby minimizing integrity and correction data for the entire service region and regarding all GNSS satellites, so the messages are publicly known schedule of message broadcasting. The service region is chosen one out of up to 64 types of message, which is only two times longer than a C/A code symbols. Each SBAS message has 250 bits and lasts for 20ms per symbol. As defined in [28], each SBAS message has 250 bits and lasts for only one second. These 250 bits are rate \( \frac{3}{2} \)-convolutional encoded from 500 symbols. When the receiver decodes the message, it needs to correctly interpret all 500 symbols per second (or 2ms per symbol), which is only two times longer than a C/A code (1023 chips, 1ms). Second, the error-checking code in the SBAS message is a 24-bit CRC (Cyclic redundant check) parity code, which can detect burst and random errors with a false-positive probability rate less than \( 5.96 \times 10^{-8} \). This parity code raises the challenge for the spoofer to correctly predict every single bit of each SBAS message. The broadcast sequence of message types for a SBAS satellite is not fixed. At every second, the SBAS service provider chooses one out of up to 64 types of messages [14] to be sent, listed in Table 1. Other GNSS satellites have a fixed and publicly known schedule of message broadcasting. Fourth, the message content in SBAS varies rapidly over time, while the messages of other GNSS systems usually stay the same for 2-4 hours. SBAS messages are designed to broadcast both integrity and correction data for the entire service region and regarding all GNSS satellites, so it is desired to not broadcast redundant messages, thereby minimizing the required data rate. However, the GNSS message is dedicated to the sender satellite, and necessary for users’ algorithms, so it must be repeat over time.

When there is at least one unpolluted channel under spoofing attack, \( x \neq x' \neq x \) will hold, and all range residuals are non-zero. We utilize Figure 6 to illustrate this concept, where the pseudorange \( \rho \) is a special form of timing information describing the satellite and receiver’s geometric relationship, and the pseudorange estimate \( \rho^* \) is a calculated value based on the estimated positions of the receiver and satellites. The range residual \( w \) is used to capture this discrepancy. In Case 3, both \( w^s \) and \( w^b \) are non-zero, while in Cases 1 and 2 all range residuals are zero. Therefore, the spoofer must have all the spoofed “circles” intersect at a single position, so that the timing information for the spoofed channels is consistent and will not be captured by RAIM models.

In our work, we obtain the pseudorange estimate \( \rho^* \) in a similar but simpler way, in order to avoid the linearization errors in (6). Let \( v \) denote the pseudorange residual used in this paper, and it is calculated as follows:

\[
\mathbf{v} = \mathbf{\rho} - \mathbf{\rho}^* = \mathbf{\rho} - (\sqrt{(x^s - x^b)^2 + (y^s - y^b)^2 + (z^s - z^b)^2} + C \cdot \delta_{tc}^t + C \cdot \delta_{io}^t + C \cdot \delta_{tr}^t - C \cdot \delta_{sc}^t),
\]

where \( \mathbf{\rho}^* \) is estimated based on the pseudorange measurement model (2), and \( x^s, y^s, z^s \) refer to satellite positions for the corresponding channels.

### 4 SBAS MESSAGE CONSISTENCY ANALYSES

SBAS systems, such as the WAAS (Wide Area Augmentation System) and the EGNOS (European Geostationary Navigation Overlay Service), broadcast GPS-like ranging signals of L1 frequency along with a 1.023Mbps PRN code. The receiver’s acquisition and tracking functions are same as that for GPS L1 signal. However, the message characteristics of SBAS signal are different from GPS L1 data bits in terms of the symbol rate, error-checking code, broadcast sequence of message types, and contents. These four factors make the SBAS messages very difficult to predict.
To gain a further intuition on the characteristics of SBAS messages, we download and analyze the decoded messages from the EGNOS Message Server (EMS) [29] hosted by the European Space Agency. The EMS contains 24-hour messages for PRNs 120, 122, 124, 126, 131, and 136, collected and post-uploaded (at least one-day later) by the base station receiver since Jan 1st, 2006. Each file in the EMS contains one-hour worth of messages, each row in which contains one message received at a time instance, being formatted as shown in Figure 7. The message content is given as an array of 63 hexadecimal numbers to represent the 250 bits in the actual broadcast message. Specifically, and without loss of generality, we analyze an EGNOS satellite (PRN 120) on a recent date – the 160th day in 2017, as well as a WAAS satellite (PRN 122, retired) in an arbitrary date in the past – the 13th day in 2006.

Table 1: SBAS message description

<table>
<thead>
<tr>
<th>Type</th>
<th>Contents</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PRN mask assignments</td>
<td>210 bits to indicate if data is provided for the satellites of corresponding PRNs</td>
</tr>
<tr>
<td>2, 3, 4, 5</td>
<td>Fast corrections</td>
<td>Pseudorange corrections (PRC) for up to 51 GNSS satellites</td>
</tr>
<tr>
<td>6</td>
<td>Integrity information</td>
<td>User Differential Range Error Indicator (UDREI) for up to 51 GNSS satellites</td>
</tr>
<tr>
<td>7</td>
<td>Fast correction degradation factor</td>
<td>UDRE degradation factor indicators for up to 51 GNSS satellites</td>
</tr>
<tr>
<td>9</td>
<td>GEO navigation message</td>
<td>ECEF coordinates and velocity of this SBAS satellite</td>
</tr>
<tr>
<td>10</td>
<td>Degradation parameters</td>
<td>Pseudorange corrections degradation factor indicators for up to 51 GNSS satellites</td>
</tr>
<tr>
<td>12</td>
<td>SBAS network time/UTC/GLONASS offset parameters</td>
<td>Time and offsets between different systems associated with the beginning of the message</td>
</tr>
<tr>
<td>17</td>
<td>GEO satellite almanacs</td>
<td>Location, velocity, and health status of the identified SBAS</td>
</tr>
<tr>
<td>18</td>
<td>Ionospheric grid point masks</td>
<td>Vertical delay estimates at specified Ionospheric grid points (IGPs)</td>
</tr>
<tr>
<td>24</td>
<td>Mixed fast corrections/long term satellite correction parameters</td>
<td>PRC, velocity, and clock correction parameters for some satellites</td>
</tr>
<tr>
<td>25</td>
<td>Long term satellite error corrections</td>
<td>Velocity and clock correction parameters for an identified GNSS satellite</td>
</tr>
<tr>
<td>26</td>
<td>Ionospheric delay corrections</td>
<td>Vertical delay estimates at IGPs at a band</td>
</tr>
<tr>
<td>27</td>
<td>SBAS service message</td>
<td>Change of UDRE in identified up to 5 geographic regions, to be used in the integrity monitoring algorithm of users</td>
</tr>
<tr>
<td>28</td>
<td>Clock-ephemeris covariance matrix message</td>
<td>Relative covariance matrix of clock and ephemeris errors for two satellites</td>
</tr>
<tr>
<td>62</td>
<td>Internal Test Message</td>
<td>For internal testing purposes</td>
</tr>
<tr>
<td>63</td>
<td>Null message</td>
<td>Filler message if no other message to be broadcast</td>
</tr>
</tbody>
</table>

Message broadcast schedule is analyzed in Figures 8 and 9. The occurrence of each type of message is plotted in Figure 8. It is observed that the two satellites have different distributions of message types. The majority of time, they both broadcast range correction data for GNSS satellites (types 2, 3, 4, 24, 25, 26). These message contents are a function of time, as they account for the ionospheric delay, tropospheric delay, and other signal propagation errors for every GNSS satellite. These message contents are also difficult to predict, because they are usually generated from real measurements by many base stations. Prediction requires highly accurate models: for example, the range correction data (PRC) in types 2, 3, 4, 5 have a 0.125-meter resolution. The sending sequence of message types is analyzed in Figure 9, where it plots intervals of each message type, i.e. the average number of messages elapsed between two appearances of the same message type (the content may be different). The WAAS satellite (PRN 122) sends type 2, 3, 24 messages every 6 messages (or seconds, equivalently), while the EGNOS satellite sends type 2, 3, 4 messages every 4 messages. These range correction messages are never interrupted by other messages and their average interval is a natural number. The intervals of other types of messages are fractional, indicating that their schedule may not be completely constant, and instead subject to interruption by messages of higher precedence.
Changes in SBAS message contents are analyzed in Figures 10 and 11. Figure 10 plots each message type’s repetition, which is the percentage of messages that are identical to their previous instance. For example, among all messages that the WAAS satellite broadcasts in the day, 53% of type-1 messages are the same as their previous type-1 messages. Types-1, 7, 9, 10, 27, 63 messages have 20–30% of repetitions, which is consistent with the slow-changing nature of their contents. For the remainder of the message types, every SBAS message appears different (0% repetition) from its previous instance, due to the nature range corrections this variance over time is expected. Figure 11 analyzes each message type’s similarity, which refers to the percentage of nibbles (a nibble is binary bits, which corresponds to a single hex digit) in a message that are not changed in one day. It is worth noting that analysis on nibble-level does not perfectly align with the fields in each message type, leading to a failure to capture the effect of some fields which may remain constant. Nibble-level analysis does offer a message-type agnostic metric, which also expedites processing as the EMS server natively stores the hex-encoded payload of each message. For the WAAS satellite, 86% of type-1, 7, 8, 10, 63 message contents remain constant throughout the day. The EGNOS satellite also has these highly similar message types. However, corrections-related message types have very low similarity.
Through these analyses, we can conclude that the SBAS message contents have few redundancies, especially for those most frequently broadcasted message types (containing corrections data; types 2, 3, 4, 24, 25, 26). Each SBAS satellite has certain rules to decide the broadcast sequence of message types, but it is not yet clear now. Because of the low redundancy and uncertain broadcast schedule, it is difficult for a spoofer to correctly predict the SBAS message.

5 EXPERIMENTAL ANALYSIS

Two sets of experiments are carried out to exhibit the change of pseudorange residuals $\nu$ both before and while the timing anomaly is present. In the first set of experiments, we inject errors into pseudoranges of the authentic signal process, to simulate the timing anomaly induced by the spoofer. In the second set of experiments, we adopt the TEXBAT spoofing binary sample dataset [4] to validate the proposed concept.

We first used the SoftGNSS software-defined radio (SDR) and the binary samples given in [30] for simulation. The results are shown in Figure 12, where 6 channels of signals are tracked in a duration of 45 seconds. Pseudoranges, solutions, tropospheric delay estimations, and other observables are reported two times per second. Starting at the 45th report of observations, we inject a slowly increasing error to channels 1 through 5 but leave channel 6 as normal at all times, at an increase rate of 2 meters/half second to mimic shifting of the code phase in a low-and-slow timing attack. The pseudorange residuals $\nu$ are calculated as in (8) and plotted in Figure 12. Ionospheric delay estimations are not factored in since in this short period, the parameters for the ionospheric model are not available. In the first 45 measurements, the pseudorange residuals on all channels are relatively constant. Note that in theory, they should all be zero-mean, but they are non-zero-mean in this experiment, likely due to the consideration of ionospheric delays. Since the anomaly starts at the 45th measurement, pseudorange residuals of channels 1, 3, 5, 6 begin to deviate from their original means. Pseudorange residuals of channels 2, 4 do not vary too much. In Figure 13, we plot the sum of squares of the deviation from $\nu$ to their means in the first 45 measurements. This result illustrates how this indicator can reveal the timing anomalies timely and effectively.
However, this simulation experiment does not restrict the anomalous channels so that they are consistent with each other. A sophisticated attacker would ensure the spoofed channels are consistent to avoid basic RAIM verification. Therefore, we turn to experimentation with the TEXBAT sample sets, which contain the signal samples as mixtures of authentic and spoofing signals. Regarded as the state-of-the-art spoofing dataset, detailed descriptions about the TEXBAT samples can be found in [4]. In order to evaluate how a typical GPS receiver might respond to the dataset, our experimental system utilizes a U-Blox M8T receiver as the target victim. The TEXBAT samples are first converted from the ishort format to gr_complex format in GNURadio [27]. After conversion, a USRP N200 software defined radio frontend is used to convert the data to an analog signal, which is supplied directly to U-Blox receiver via cable. U-Center, free software provided by U-Blox, is used to generate an output file which is converted and post-processed by our UBX parser and RTKLib [26]. RTKLib extracts the navigation messages to generate a RINEX file. Our UBX parser processes UBX messages, a proprietary protocol developed by U-Blox for use with their receiver. The UBX messages used and their purposes are described in the Figure 14. In our postprocessing program we read all input files, associate observables with their time stamps, and finally calculate the pseudorange residuals.

Two scenarios in the TEXBAT sample set are processed: the clean static scenario – only authentic signals, and the 4th scenario – static matched-power position push [4]. In both scenarios, the U-Blox receiver is able to receive the SBAS L1 signals from PRNs 133, 135, 138. However, we disallow the SBAS signals to be used in the navigation engine. The experimental process was not as smooth as could be expected. When the GPSDO (GPS disciplined oscillator) is enabled for the USRP N200, which serves to provide a more accurate and stable internal clock, to broadcast TEXBAT samples the U-Blox receiver cannot replicate the GPS traces reported in [4]. Instead, the receiver’s trace for the scenario 4 appears normal (i.e. unspoofed), as shown in Figure 15. When we disable the GPSDO on the USRP N200, the receiver resolved GPS traces as expected (see Figure 16). As of the writing of this paper the reason for this difference is not fully understood. It may be related to the built-in anti-spoofing module of the U-Blox M8T receiver. In order to disable the anti-spoofing feature the receiver must be limited to a single GNSS constellation, which would mean disabling SBAS support and is thus not an option. Therefore conducting experiments with the GPSDO disabled is the best option. As shown in Figure 16, corresponding to scenario 4, the spooper begins to mislead the victim’s position in the 110th – 120th seconds following the time of the victim’s first position fix. The maximal position deviation...
induced by the spoofer is about 400 meters in the entire period. For portions of this period, the receiver is not able to output position solutions.

Figure 17 shows the pseudorange residuals of three SBAS L1 signals, where no data is reported if the receiver does not have a position fix. The pseudorange residual values are non-zero, because the reported receiver clock offset has been found constantly away from its true value in U-Blox’s protocol for unknown reasons [31]. The range residuals are reduced by approximately 150 meters when the victim is spoofed from the original position to a new position. This discrepancy suggests the spoofing attack can be detected. In the meantime, GPS channels are self-consistent as shown in Figure 19, as intended and designed by the spoofer.

For comparison, the results for the clean static scenario are plotted in Figures 19 and 20. In this scenario, all signals are authentic so the SBAS channels are consistent. Their range residuals do not show discrepancies according to Figure 20.
6. SUMMARY

GNSS spoofing is increasingly considered an emerging threat to many applications due to the open-specification nature of the GNSS signals. Live spoofing of GNSS signals has been demonstrated on multiple occasions, and there is no solution that can guarantee detection of GNSS spoofing. To address this important issue, in this paper we analyze the resource requirements for spoofers to launch spoofing attacks. Knowing that the resource requirement is proportional to the quality of spoofing attacks, we believe that it is very difficult for spoofers to fully simulate all GNSS constellations, especially SBAS channels, simultaneously. Off-the-shelf receivers can readily observe a large number of GNSS channels, including SBAS signals. We also propose two simple consistency-checking models for detection of RMT based GNSS spoofing by exploring the abundant GNSS signals. We show that the RAIM algorithm can be readily tailored for detection of timing inconsistencies, and it is highly challenging to spoof the dynamic content of the SBAS messages. Together, these observations can significantly improve our ability in spoofing detection. Experiments demonstrate the practicality of these techniques.

REFERENCES


