International Symposium on GNSS 2015
Kyoto, Japan.
November 16-19, 2015.

Detection and Exclusion RAIM Algorithm against Spoofing/Replaying Attacks

Kewei Zhang, Rashedul Amin Tuhin and Panos Papadimitratos*

Abstract Malicious attacks, notably spoofing and replaying, in addition to jamming, have been shown increasingly practical even for attackers without high levels of sophistication. This makes malicious interference and manipulation a significant threat for Global Navigation Satellite Systems (GNSS)-based applications. Receiver Autonomous Integrity Monitoring (RAIM) has been widely used in safety-critical navigation applications to ensure the integrity and reliability of the navigation function, without being designed with malicious faults in mind. Along the lines of the RAIM approach, here we investigate an alternative method to exclude faulty pseudo-range measurements and obtain an as accurate as possible position estimate. We propose the use of an approach that allows the receiver to classify position estimates into, roughly speaking, those that are faulty (affected by an attacker) and those that are not. Identifying an intuitive match to GNSS-based positioning, we propose the use of a clustering algorithm for this classification. We find that our so-called Clustering-RAIM (CRAIM) is simple to configure and efficient in terms of computation, and our evaluation, simulating adversarial effect, shows it can be effective without any prior assumption on the number of faulty measurements; as long as they are the minority of the available ones.

Keyword Clustering, Multiple outliers, RAIM, Spoofing, Replaying

1. Introduction

Global Navigation Satellite Systems (GNSS) have enabled a wide gamut of civilian applications, many of critical importance, ranging from positioning of valuable assets, vehicles, and individuals to time synchronization of information systems. The importance of precise positioning (and time synchronization) and the high stakes for many applications expose to significant threats. The nature of GNSS, with relatively easy to overpower signals and no security measures for civilian applications readily available, makes a range of attacks possible, even without exotic equipment. Jamming attacks, i.e., intentional interference that prevents reception, have been shown possible with simple equipment and nowadays several commercially available receivers provide improved resilience. Spoofing attacks, i.e., forging of signals that alter the victim receiver's position (and navigation solution overall), have recently been shown relatively easier to mount. Relaying attacks [1], with the adversary recording and retransmitting GNSS signals, can also manipulate the victim's position by altering the estimated pseudo-ranges even if cryptographic protection (authentication), in upcoming systems, were available.

All these attacks essentially harm the integrity of the GNSS-based positioning. Considering the Global Positioning System (GPS) in particular, integrity has been long identified as an important issue. However, this has been done at first with benign faults (errors) in mind, e.g., signal propagation impairments or unintentional interference. The Receiver Autonomous Integrity Monitoring (RAIM) technology was proposed exactly to assess the integrity of GPS signals, the basic idea being to use redundant satellite measurements [2]: needing four satellites signals to obtain a navigation solution but overall having several more satellites visible, RAIM, simply put, obtains one solution and checks the error between the resultant estimated pseudo-ranges and the measured ones, for all visible satellites. Comparing this to a predefined threshold, RAIM seeks to detect, in the form of a hypothesis test, the erroneous signal as the one with the highest error. More recent works, discussed in Sec. 2, have investigated and proposed RAIM variants that assume multiple faulty signals.

Intuitively, RAIM can be seen as agnostic to the nature of the error: one or more outlier pseudo-ranges can be caused by malicious faults, i.e., the result of an attack (adversarial behavior). This motivated a number of works that investigate RAIM as a countermeasure against spoofing and jamming attacks [5]. These and follow-up works, discussed in Sec. 2, essentially extend conventional RAIM and seek, initially, to detect a single offending signal, i.e., a single measurement manipulated by an adversary. The resultant schemes either inherit, in a sense, the challenges of the conventional RAIM, notably the need to identify appropriate thresholds, or they resort to more involved models and complex computations.

We are interested in a simple and efficient method to classify position solutions, based on different subsets of available pseudo range measurements, as ‘outlier’ or ‘inlier’ - implying that an inlier position estimate includes only legitimate measurements; while, inversely, an outlier includes one or more measurement manipulated by an adversary. We are after a method that relies on a simple assumption on and model of the adversarial strength, and simple to set parameters. The basic notion of GNSS-based positioning is that any position estimation that does not include any faulty measurement will be close to the actual one, as well as the position calculated based on subsets of legitimate measurements. This is what

* K. Zhang and P. Papadimitratos are with the Networked Systems group at KTH (Royal Institute of Technology). R. A. Tuhin contributed to this work while at KTH.
Web: www.ee.kth.se/nss
Email: {kewei, ratuhin, papadim}@kth.se

intuitively motivates us to propose and investigate the use of a *clustering algorithm* [34] to classify all possible position estimates. The novelty of this work lies in proposing and investigating this alternative approach in line with the RAIM approach but originating in a different area (data bases), for classifying position estimates and eventually sifting measurements to exclude the faulty (adversarial) ones.

The resultant proposed here Clustering-RAIM (CRAIM) algorithm can detect the existence of multiple manipulated signals simultaneously, by checking the receiver's position consistence iteratively and then returning the average position based on all the subsets of deemed legitimate signals (after excluding the adversarial ones). The underlying assumption is that the minority of the received signals are affected by the adversary.

The paper starts with brief explanation of conventional RAIM, followed by a concise survey of more recent work for multiple outliers, as well as other GNSS security approaches (which could work in parallel with CRAIM or as complement to each other). Then, the adversary model is presented before presented the proposed Clustering-RAIM. Finally, we illustrate the operation and simulation results based on on-line observation file and navigation file.

2. Related Work

2.1 Conventional RAIM

Three methods for GPS integrity mainly attracted attention: (i) range-comparison, (ii) least-squares and (iii) parity methods. All three methods are equivalent, yielding identical results eventually [3]. Assuming there are five satellites in view, the range-comparison method solves the four equations with four satellites, and the obtained solution predicts the fifth measurement. Then, it compares to the actual, measured value. There is distinction for range and position comparison [9]. For range comparison, the receiver estimates its position with four satellites and then predicts other measurements. The predicted values will be compared with the actual measured values to verify the measurements consistence. After that, the receiver detects an outlier by comparing the bias between the predicted and the measured pseudo-range with respect to a predefined threshold. For the position comparison method, the receiver estimates its position with all five satellites and with all subsets of four satellites, and then the differences between the all-satellites solution and each of the four satellites solutions is considered as a bias to detect an outlier.

The Least-Squares method predicts all measurements and all errors for each satellite and then it sums the squares of all residual errors to compare with a threshold and thus detect whether an outlier exists. The linear measurement model for the receiver could be written as:

$$y = Gx + \xi \tag{1}$$

where $\boldsymbol{x}$ is receiver position and clock bias vector, $\boldsymbol{y}$ is the pseudo-range measurements, $\boldsymbol{G}$ is the observation matrix and ξ is the error vector. The goal is to find the residual errors between the measured ranges e (Eq. (2)) and the predicted ones, $\hat{y}$, (Eq. (3)):

$$e = y - \hat{y} \tag{2}$$

$$\hat{y} = G(\mathrm{G}^T\mathrm{G})^{-1}G^T y \tag{3}$$

The sum of squares (*SSE*) of the range residual errors is defined as a test statistic of detection failure and exclusion (FDE): $SSE = e^T e$.

The parity scheme, somewhat similar to the range-comparison method, performs a linear transformation of the measurement vector, and then it looks at the magnitude of the partitioned vector, as per Eq. (4), where *p*, called the parity vector,

$$\begin{bmatrix} \hat{x}_{LS} \\ p \end{bmatrix} = \begin{bmatrix} (\mathrm{G}^T\mathrm{G})^{-1}G^T \\ P \end{bmatrix} y \tag{4}$$

is defined as the test statistic. For a given $m \times n$ matrix G with rank n , matrix P could be found with rank $[\mathbf{P}] = m - n$, $PP^T = I_m$ and $PH = 0$ [2]. Our approach could be classified, broadly speaking, as a position-comparison method.

2.2 RAIM for Multiple Outliers and Attack Mitigation

Recently, a few methods have been proposed to monitor for multiple outliers. Most of them focus on two outliers [10], among of which [11] used $\omega - test$ that has limitations for correlated test statistics as analyzed in [13]; there is also a negative affect that good measurements could be miss-detected. [10] analyzed the performance based on an assumed specific number of faults; the number needs to be known in advance for analysis. [12] calculated position solutions based on subsets of four satellites and compares with the estimate with pseudoranges not contributing to this solution. [12] proposed a new multiple-hypothesis RAIM algorithm based on monitoring the error vector itself with several consecutive epochs, but this algorithm cannot identify errors. [14] tried to extend the situation of dealing with single failure to multiple failures. [15] focused on the theory of reliability of extending single outlier to multiple outliers, but it cannot identify the number of outliers.

As the development of clock gain and ephemeris accuracy, and the deployment of new GNSS, high demanding phase of flight is considered, especially for those requiring vertical guidance. The GPS Evolutionary Architecture Study (GEAS) outlined an Advanced RAIM concept that relies on a ground system to provide periodic updates the nominal performance and fault rates of the multiplicity of contributing constellations [16]. The Integrity Support Message (ISM) [20], containing the integrity data, is determined on the ground. It carries key parameters used by ARAIM, including clock and ephemeris errors for integrity and continuity, also probability density function of the error distribution, etc.

Several works have introduced RAIM as a countermeasure against spoofing and jamming [5], and they used the same concept as the normal RAIM, plus considering the pseudorange residual. [5] developed a GPS/INS integrated navigation system to directly detect spoofing attacks, which utilizes the redundancy of INS measurements, instead of satellites measurements redundancy. Then [6] extended the work to compute upper bounds on the integrity risk under the impact of wind gust without simulating individual aircraft. [7] introduced interacting multiple model (IMM), comparing to conventional RAIM, to detect and identify a single interference. [8] simply proposed RAIM as one of countermeasures and it combined Doppler shift frequency errors and pseudo-range errors together for a single spoofing signal.

2.3 Other GNSS Security Approaches

Here, we briefly survey other proposals in the literature seeking to protect the GNSS-based positioning and synchronization. [21] proposed approaches based on including external units or internal signal analysis, such as inertial sensors, clock offset, Doppler shift, signal relative power, cross-correlation of $L1$, $L2$ and residual signal analysis. [24] provided cryptographic protection for GNSS signals that requires modification of the system structure, for instance message authentication and signal encryption. Moreover, [27] proposed using multi-antenna as an autonomous method to protect receiver from the spoofing attack.

Those are orthogonal and can co-exist with the RAIM-based approaches and accordingly our CRAIM. Simply put, an alternative security method could for example detect an attack and then invoke CRAIM to identify the erroneous signals. Or, a countermeasure that rejects a specific signal could provide that input to CRAIM, readily excluding one (or more) outlier and then allowing CRAIM to perform computations based on the remaining measurements (pseudo-ranges).

3. Attacker Model

We do not dwell on the exact type or category or method of attack - in other words, we assume an attacker either with jamming and spoofing or jamming and relaying/replaying capabilities; the latter affecting the victim receiver even if the navigation message contents cannot be manipulated by the attacker. We do not elaborate, but we assume implicitly that the adversary is sufficiently sophisticated in order to spoof or replay signals from satellites that cannot be readily rejected by the receiver. Rather, we abstract the effect of the attacker on the GNSS-based positioning as a measurement error imposed on a pseudo-range measurement. Consider that the GNSS receiver computes a position (and clock offset) based on four satellites/measurements, as those in Eq. (5), where ρ_i is the pseudo-range measurement to the i^{th} visible satellite vehicle, SV_i, (X, Y, Z) are the receiver's coordinates, and (x_i, y_i, z_i) the coordinates of SV_i.

$$\rho_i = \sqrt{(X - x_i)^2 + (Y - y_i)^2 + (Z - z_i)^2} + \xi \qquad (5)$$

We let ξ, the measurement error, in Eq. (5), encompass the attack effect, or more precisely, capture any error, benign and malicious (or possibly their joint effect). In that sense, we remain agnostic to error cause and in line with the broader RAIM approach.

We constrain, however, the power of the adversary.

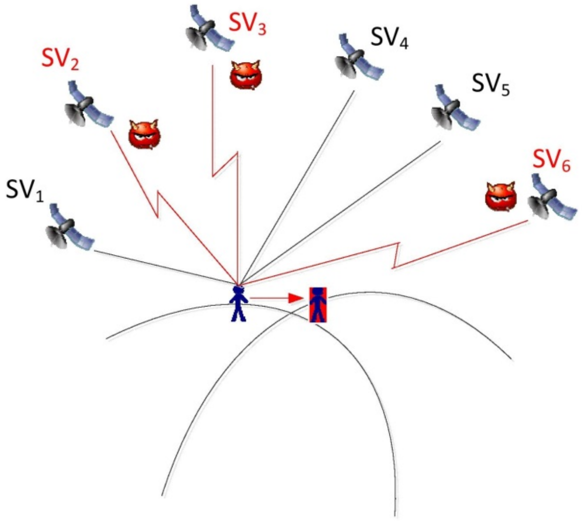


Fig. 1. Illustration on attack affecting multiple signals

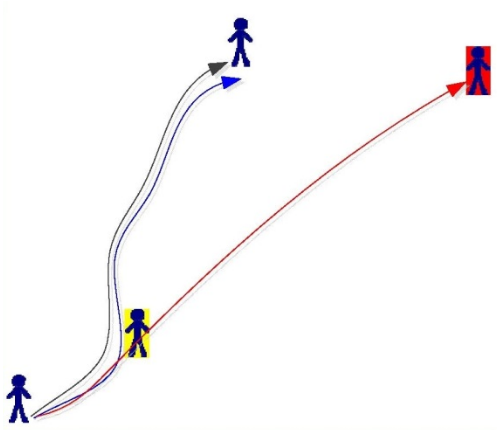


Fig. 2. Subtle attacks

Assuming at any point in time there are *N* visible satellites and thus pseudo-range measurements, we assume that no more than M of those measurements can be the result of (or more generally affected by) an attack. More specifically, we constrain this to require at most *M* < *N-5* adversarial (faulty) measurements.

This limitation on the power of the adversary is necessary because if the majority of available signals were adversarial then there can be no guarantee of computing a position that is not affected by the adversary. Still, this is a strong adversary model, assuming multiple adversarial signals. The above-mentioned assumption does not presume any method that could alone disqualify an adversarial signal (e.g., by detecting exceedingly high power).

The motivation of the adversary is the same as that for any attack, to control the victim receiver's position. Without loss of generality, Fig. 1 provides a simplified illustration with multiple attacker-controlled measurements that shift the perceived position of the victim (blue) to the attacker's intended position (red).

The victim receiver could start in hot, warm or even cold mode; this does not affect the result of our algorithm. In the worst case, the cold start mode, the receiver has no previous information about any system parameter. One can actually consider a subtle adversary, as illustrated in Fig. 2[1], which 'shifts' the victim receiver's position in relatively small steps, gradually increases the distance to the target location. Any detection tool along the lines of RAIM would most likely not detect the adversarial actions if they result in very low ξ and thus very low victim displacements. Broadly speaking, the exact effect of such attack dependents on number of satellites and their geometry, the algorithm will detect and identify them. The exact strategy of an attacker is beyond the scope of this paper.

4. Clustering Receiver-Autonomous Integrity Monitoring (CRAIM)

4.1 Background

In data mining, several methods have been developed to detect multiple outliers [29] within a data set. Among those, clustering is a methodology to identify multiple outliers in linear regression [30]. Clustering, strictly speaking not an algorithm but rather is a task/technique for grouping similar data samples, cannot be readily identified as having an optimal instantiation or algorithm [31]: this is so because the success of clustering highly dependents on the data and the purpose of the investigation.

Cluster analysis takes a set of *n* observations for *p* variables, then it calculates the similarity between these observations and finally groups some observations into one cluster based on inter-observation similarities. [32] pointed out that Euclidean distance is the mostly used method to measure the similarity because of its simplicity and efficiency. [33] surveyed several clustering algorithms, among of which density-based clustering is chosen in our work that groups the closest samples (high density) to inlier and makes these points with low density as outlier.

Density-based Spatial Clustering of Applications with Noise (DBSCAN) [34] is a clustering method that, as illustrated in Fig. 3, if we define the cluster size $N=7$ and the radius as ε , then the blue points are grouped together. However, if we define $N=8$ for the same radius, ε , the method will fail finding such cluster. When define $N=8$ and the radius as R, the green point will be added to the group, then the red points will be marked as outliers.

4.2 Our Approach

Our proposal is essentially to combine RAIM with the DBSCAN algorithm; this results in the proposed Clustering-RAIM (CRAIM), an algorithm to detect and identify offending (interfering, spoofed or replayed) signals and the corresponding pseudo-range measurements at the GNSS receivers. The approach is simple to implement, not requiring any receiver equipment modification.

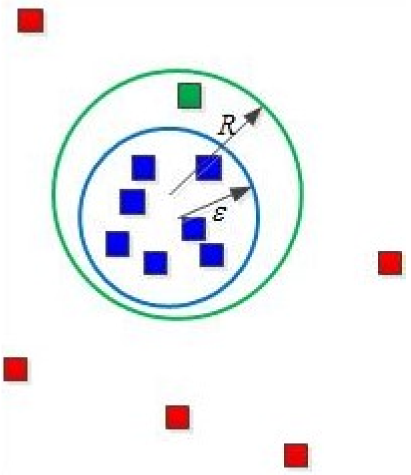


Fig. 3 DBSCAN algorithm illustration.

The algorithm input is: (i) all available position estimation based on all possible combinations of available pseudorange measurements; (ii) cluster radius ε (iii) number of position estimates in a cluster, MinPoints.

The DBSCAN clustering algorithm, which is efficient even for large data sets, operates on a relatively small number of possible position estimates and it has low computation cost. The parameters needed for the cluster can be set based on the GNSS positioning notions and simple heuristics. For different number of satellites, the MinPoints will be set according to number of combinations, shown in Fig. 4. Those neighboring data points will be close to each other, i.e., densely so, based on expected benign inaccuracies of positioning. This implies one can set a relatively small ε . As per the original paper [34], we use a k-distance approach, i.e., the distance to the k nearest neighbors to compute jointly ε and MinPoints, which, as per the heuristic and DBSCAN intuition, corresponds to the so-called thinnest cluster DBSCAN outputs. To obtain values for those parameters, we leverage in Sec. 5 the used data and calculate indicative values without any manipulated measurements.

Assuming that out of N visible/available signals/satellites no more than M are adversarial (or erroneous), the steps of CRAIM are:

1) Start by assuming there is one manipulated satellite, i.e., that $M=1$; the legitimate satellites are N-M=N-1.
2) Calculate the position from $K=N-2$ satellites, because there will be only one 'good' position result if $K=N-1$ thus making clustering meaningless. The cluster size is C_{N-M}^{K} .
3) Define a CRAIM threshold, *radius* ε , which indicates the sought level of accuracy.
4) Based on the aforementioned threshold, use DBSCAN to find whether a cluster of size C_{N-M}^{K} exists or not.

[1] The black line is the actual trajectory, the adversary-induced/'shifted' trajectory to the blue line (meters or ten meters difference) that probably cannot be detected, yet does not affect the receiver for most applications. However, if the adversary tries to misguide the GNSS receiver to the red line, all the manipulated signals would be detected when the receiver reaches at yellow spot.

5) If not, decrease *K* by 1, and iterate from step 2).[2]
6) If the results are not conclusive (no cluster of specific size is found), reset *M* by increasing 1, and iterate from step 1).
7) If the result is conclusive and a cluster is found, then CRAIM outputs to the receiver that there are *M* satellites being manipulated.
8) Then calculate the final position with the remaining *N-M* signals.

Intuitively, the algorithm attempts determining for each iteration whether a cluster of position estimates can be formed based on a set of measurements with a progressively reduced cardinality. If a cluster of an attempted size cannot be formed, then it surmised that one or more of the used measurements is (are) faulty. By increasing the presumed number of faulty signals, M, in fact by iterating on this number, the algorithm successively 'removes' a different subset of M signals and tests if a cluster can be formed for position estimates based on the remaining N-M signals. If so, they are deemed a set of non-faulty measurements.

Of course, it is important to note that if the manipulation by the attacker is subtle, within margins that result in position estimates with in the cluster radius, then CRAIM does not exclude those. But such limited perturbation implies the attack or fault remains imperceptible and can only cause a mild increase in the inaccuracy of the position. The usefulness (if any) of such attacks and the possibility to thwart them are not investigated here.

At the conclusion of the above-mentioned algorithm, the number and the indices of the deemed faulty measurements (and thus satellite signals) is determined. Having such as set, based on the position results that produce the cluster exclude the adversarial signals and a new position can be calculated. This will be better explained by an example in next section. The basic idea for exclusion is that the algorithm finds all satellites that are used to obtain those position results being grouped together, then examines which satellites contribute to those position results, the rest are marked as malicious.

In terms of actual implementation, there is no need to calculate the cluster size in real time, i.e., there is no need to expend the power and undergo the processing delay, even though small. The CRAIM running GNSS receiver can pre-compute and store these values for the attempted/tested cluster sizes in its memory (as illustrated by the Pascal Triangle in Fig. 4). At each iteration, the receiver uses values it needs, indicated by the red line in the figure, for which the total number of visible satellites/available signals is *N*.

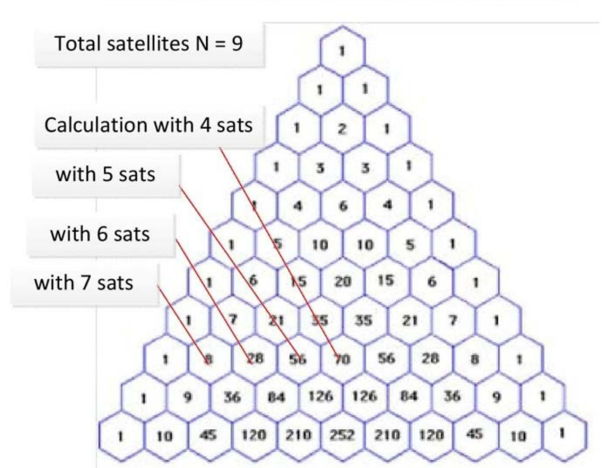


Fig. 4 Possible position calculations with different number of satellites

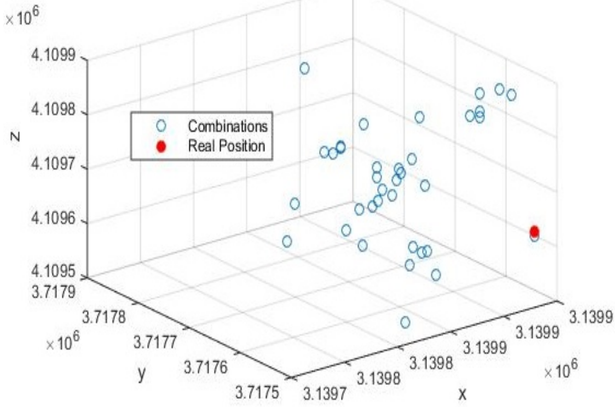


Fig. 5 Position calculations assuming only one adversarial satellite signal, calculated with different subsets of 7 satellites

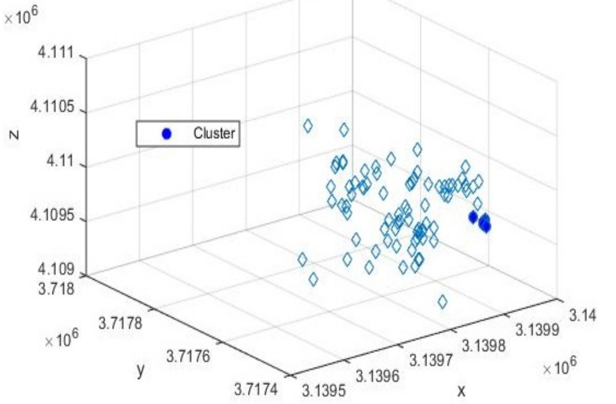


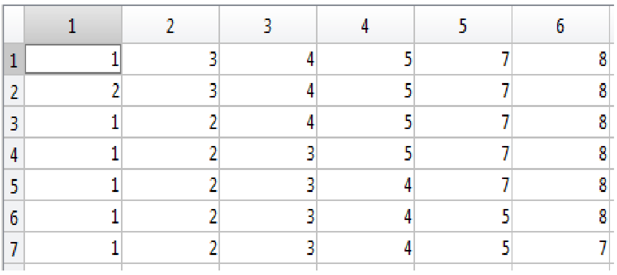
Fig. 6 Position calculations from subsets of 6 satellites, assuming two adversarial satellites signal.

5. Algorithm Evaluation

Evaluation setup: The GNSS data, that is, the navigation file and the observation file, are downloaded from [35]. In those, there are *N=9* visible GPS satellites in total, with *PRN* {2 5 6 31 10 21 12 25 29}. To emulate the attack effect, we manipulate *M=2* satellites, whose pseudo-ranges are modified (Recall Eq. (5)). In the simulation, we increased the pseudo-ranges of two adversarial satellites by 150m, 300m (which is about 1 μs delay), 600m, and 900m. Again, this is a protocol selectable value that represents. This value is

[2] It is more likely that it is not necessary to iterate, because satellites with good geometry (low dilution of position) will obtain very similar results when using, for example 6 or 7 of them

conservative, it might allow subtle attacks to have a signal with a small adversarial perturbation (low value ξ) clustered with legitimate ones. But given the small effect, the position accuracy would not be severely affect. On the other hand, if we set $\mathcal{E}$ to a lower value, CRAIM might miss-detect a legitimate signal, leaving outside the cluster. Inversely, if we set $\mathcal{E}$ to a large value, the algorithm would lose the ability to detect the manipulated satellites.



	1	2	3	4	5	6
1	1	3	4	5	7	8
2	2	3	4	5	7	8
3	1	2	4	5	7	8
4	1	2	3	5	7	8
5	1	2	3	4	7	8
6	1	2	3	4	5	8
7	1	2	3	4	5	7

Fig. 7 Indices of all satellites contributed to the cluster

CRAIM operation: First, the CRAIM receiver assumes the number of adversarial satellites *M=1*, thus assuming eight legitimate satellites. The goal is for CRAIM to find those eight satellites. Thus, it searches a cluster with DBSCAN with a receiver defined $\mathcal{E}$, corresponding to the following sets of points (position solutions): $\{C_8^8 = 1, C_8^7 = 8, C_8^6 = 28, C_8^5 = 56, C_8^4 = 70\}$ (See the Pascal Triangle in Fig. 4). For each value, the algorithm runs DBSCAN to check whether it can find a cluster containing that number of points. Clearly $C_8^8 = 1$ is meaningless for clustering, thus not considered.

The receiver needs at least four satellites to calculate its position, therefore C_8^4 is the biggest cluster that the algorithm needs to search. However, after this process, the algorithm fails finding a cluster with the aforementioned number of points; therefore it believes there is more than one adversarial satellite. Fig. 5 shows the position results calculated using seven satellites, in which case the cluster size is eight. However, the algorithm cannot find a cluster containing eight position results with the defined radius $\mathcal{E}$. We can see that there is only one point close to the real position, which is obtained with seven legitimate satellites.

Second, the receiver will reset the number of adversarial satellites *M=2*, search the cluster size with the number of following points: $\{C_7^7 = 1, C_7^6 = 7, C_7^5 = 21, C_7^4 = 35\}$. If the algorithm still cannot find one cluster, it believes that more than 2 manipulated satellites exist. Then, it will increase *D* by 1, and redo the same process until it finds a cluster. In this case, the receiver finds one cluster, and accordingly it believes two satellite signals are modified, as shown in Fig. 6. The figure shows that the algorithm can find a cluster containing seven position results with defined radius, marked with blue color. In fact, in order to optimize the algorithm, it is better to find a cluster with position points calculated with more satellites that is $C_8^7 = 8, C_7^6 = 7, C_6^5 = 6$ for each iteration. The reason is that the position will be more accurate when calculate with more satellites.

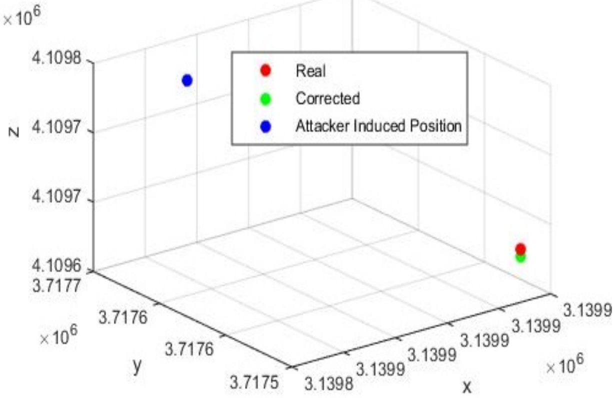


Fig. 8 Receiver position: real one (red), shifted one (blue) and corrected one (green)

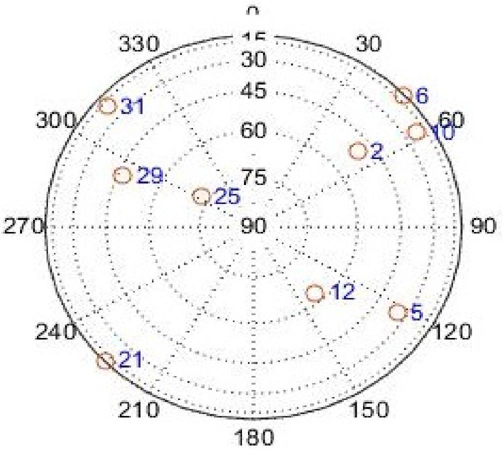
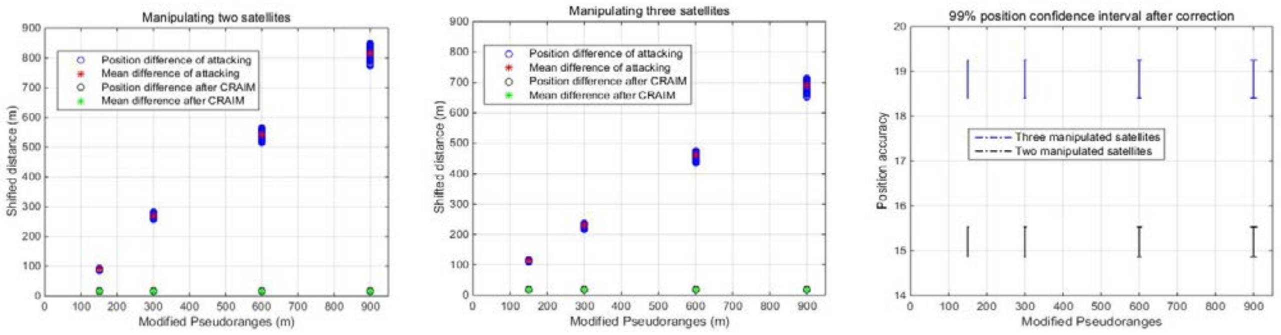


Fig. 9 Constellation of satellites in the simulation

Third, after determining the number of adversarial satellites, CRAIM needs to identify those out of all available satellites. In the last step, the receiver has the information of these satellites used to calculate the positions that have been grouped as a cluster. The indices of all the satellites are shown in Fig. 7, which is generated when the cluster size is seven. Each row contains the index of six satellites, which are used to calculate one position result, and there are totally seven position results contributing to this cluster. Then, the receiver obtains all unique values from all the indices, which would be *{1, 2, 3, 4, 5, 7, 8}* that are satellite indices of the related legitimate signals. Excluding these satellites from the set of available ones, the indices of adversarial satellites could be identified as *{6, 9}*, corresponding to *PRN {21 29}*.

Simulation results: After identifying and excluding the outliers, the receiver will calculate its position only based on the remaining satellite signals, as shown in Fig. 8. The distance between the actual position (in red) and the attack-induced position (in blue) is 173.22 m; the distance between the actual



(a) (b) (c)

Figure. 10 Attacker gradually changes the pseudorange: (a) two manipulated satellites, (b) three manipulated satellites, (c) position confidence interval after correction

position and CRAIM-corrected position (in green) is only 5.01 m, which is good enough for most applications.

Another parameter that affects the result is the constellation of available satellites; Fig. 9 provides the constellation of satellites in our simulation for one snapshot. We can tell that if we remove satellites *{5 12 21}*, all other satellites remaining in the upper part of the illustrated constellation give a high geometric dilution of precision (GDOP). Then, after running CRAIM, we can only detect and identify satellites *{5 12}* as being manipulated at once. However, if satellites *{2 12 29}* are manipulated, the algorithm works perfectly to identify all of them simultaneously, because the remaining satellites were in favorable constellation positions. An intelligent attacker would choose to manipulate several satellites to lower the GDOP of remaining satellites, which would then increase the workload and prolong the processing time for the receiver until it manages to exclude all of them. Finally, we investigated an attacker that gradually shifts the receiver from *150* m - *900* m, notably the performance of the receiver, in fact the positioning accuracy. With the same observation and navigation data, we consider 15 minutes of observation data with epoch interval of 30 seconds. Within these 15 minutes, the adversary modifies with {150, 300, 600, 900} m separately. Based on the heuristic result after analyzing the database with k-dist function, we find that it can cluster all position calculations with subsets of 7 satellites when ε is larger than 37 m and it can do so with subsets of 6 satellites when ε is larger than 70 m. So we choose 50 m and 80 m for two cases in this simulation: two and three manipulated satellites. The results are presented in Fig. 10(a) and Fig. 10(b) respectively. We can see that CRAIM can guarantee the receiver's position accuracy to be about 15 m with 99% confidence interval for two manipulated satellites and about 19 m with 99% confidence interval for three manipulated satellites, as shown in Fig. 10(c), no matter how the adversary gradually changes the pseudoranges.

6. Discussion and Conclusion

The paper presented an approach based on RAIM and clustering algorithms as a countermeasure against spoofing and replaying attacks. The simulation results show that it can detect and exclude multiple adversarial satellites signals simultaneously, and then obtain a new position with the legitimate signals. The algorithm has a certain limitation that it can detect and exclude outliers simultaneously with at least six satellites in view, because with five satellites it can only detect an outlier, but not identify it; which is also the limitation of all RAIM algorithms. Another limitation is that it can detect at most *(N-4-1)* faults at once, which could be a problem when operating with a single GNSS constellation and in the presence of an adversary that can manipulate numerous satellites signals. However, with more GNSS satellites operational, such as Beidou, Galileo, etc., there will be around 30 available satellites at one epoch; this would allow CRAIM to thwart adversaries that introduce many manipulated satellite signals. Given other works on detecting and identifying multiple outliers with hypothesis testing, our future work will examine and compare those algorithms to CRAIM.

We sketch how to connect the clustering radius to a tolerable inaccuracy (error) by a user application in Appendix A.

Appendix A. Relation of Cluster Radius and Tolerable Error

Eq. (1) gives the observation linear model in the receiver:

$$y = Gx + \xi$$

where the error term $\xi \sim \mathrm{N}(0, \mathrm{W}^{-1})$, W^{-1} is a known nonsingular covariance matrix. When there are errors in pesudoranges measurements, the error term will be [14]

$$\xi \sim \mathrm{N}(\mathrm{e}, \mathrm{W}^{-1}) \tag{6}$$

where $\mathrm{e} = \begin{bmatrix} 0 & \cdots & e_i & 0 & \cdots & 0 & e_j & \cdots & 0 \end{bmatrix}^T$ that means the i^{th} and j^{th} pseudoranges measurements have errors. The receiver position estimation is

$$\hat{x} = (\mathrm{G}^T \mathrm{WG})^{-1} G^T W y \tag{7}$$

Then the position residuals could be written

$$d = \hat{x} - x = (\mathrm{G}^T \mathrm{WG})^{-1} G^T W \xi \tag{8}$$

Here the receiver has a distance error upper bound λ that the victim can tolerate, therefore the receiver designs

$$\begin{cases} d < \lambda & \text{no alarm} \\ d > \lambda & \text{alarm} \end{cases} \tag{9}$$

Idea of defining the radius threshold ε actually aims to

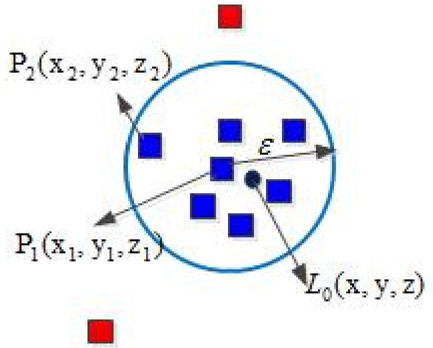


Fig. 11 Illustration of the cluster radius estimation

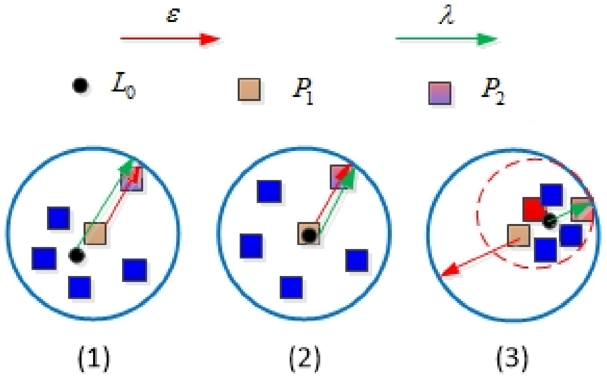


Fig. 12 Three cases for design the threshold

satisfy a distance upper bound, λ that the victim can tolerate between the real position and positions corresponding to satellite subsets. In Fig. 11, the maximum distance is the distance between L_0 and P_2, and the maximum distance should not be larger than the distance the receiver can tolerate. The acceptable perturbed position, λ is defined by the receiver, regarding to different applications.

In CRAIM, the receiver starts verifying position results of different subsets by assuming the number of manipulated signals from $D = 1$, while CRAIM tries to find the cluster with specific number of points by iterating the cluster radius, threshold ε, from 1 to ε. The process starts assuming one arbitrary point as a core point for a cluster, iterating the cluster radius. If it cannot find a required cluster, it will move to another point. If the algorithm cannot find one demanded cluster by iterating all points, it will set the number of manipulated signals to $D = 2$, and then do the iterations again, until it finds a cluster with specific requirement that makes the point, for instance P_1, as the core point.

Assuming that the cluster includes all blue points (position results calculated with different combination of satellites), and then we can calculate the final position:

$$x = \frac{\sum_i^N x_i}{N}, y = \frac{\sum_i^N y_i}{N} \text{ and } z = \frac{\sum_i^N z_i}{N} \qquad (10)$$

where N is the number of position points.

Assuming the farthest point is P_2, and then the requirement is:

$$\sqrt{(x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2} \le \lambda \qquad (11)$$

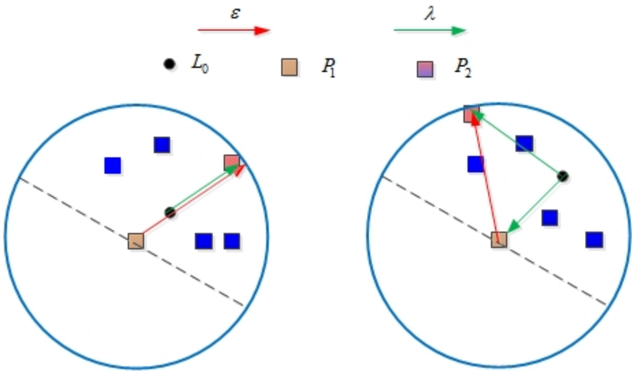


Fig. 13 Two situations of case (3)

The problem has basically three different cases, illustrated in Fig. 12:

1) L_0 and P_2 are at two sides of P_1;

2) L_0 and P_1 are at the same position;

3) L_0 and P_2 are at one side of P_1;

For the simplest case (2), we have $\varepsilon = \lambda$.

For case (1), the worst is when L_0 P_1 P_2 are at one line and $\lambda = 2\varepsilon$, so $\lambda \le 2\varepsilon \Rightarrow \varepsilon \ge \lambda / 2$.

For case (3), the farthest point from real position L_0 could be any points, and the cluster (red dashed circle) taking red point as core point has smaller radius than the cluster (blue circle). Therefore, if ε is small, when the algorithm tries to find a cluster that iterates P_1 as core point earlier than the red point, it cannot succeed until it picks the red point, which could decrease the efficiency of the algorithm. However, if ε is big, it will succeed in finding a cluster taking P_1 as core point if it verifies P_1 earlier. But another efficiency issue is that it takes more time to iterate from 1 to ε for other points earlier than P_1. There are two situations in case (3) that is illustrated in Fig. 13. If L_0 P_1 P_2 are at one line, P_2 is farther than other points, then $\varepsilon - \lambda \le \lambda \Rightarrow \varepsilon \le 2\lambda$ (the left plot of Fig. 13). If L_0 P_1 P_2 are not at one line (the right plot of Fig. 13), the largest ε will be obtained when λ is defined in advance is described in the right plot, where $\lambda^2 + \lambda^2 = \varepsilon^2 \Rightarrow \varepsilon \le \sqrt{2}\lambda$.

Combining the three cases, we can include all possible ε and we can reach a conclusion: $\lambda / 2 \le \varepsilon \le 2\lambda$, after considering all situations.

References

[1] K. Zhang, P. Papadimitratos, "GNSS Receiver Tracking Performance Analysis under Distance-Decreasing Attacks," in Proceedings of the ICL-GNSS, Gothenburg, Sweden, 2015.

[2] M. A. Sturza, "Navigation System Integrity Monitoring Using Redundant Measurements," Journal of Navigation, vol. 35, no. 4, pp. 483–501, 1988.

[3] R. G. Brown, "A baseline GPS RAIM scheme and a note on the equivalence of three RAIM methods," Journal of Navigation,

vol. 39, no. 3, pp. 301–316, 1992.

[4] F. Van Diggelen and A. Brown, "Mathematical aspects of GPS RAIM," in Proceedings of PLANS. IEEE, 1994.

[5] S. Khanafseh, N. Roshan, S. Langel, F.-C. Chan, M. Joerger, and B. Pervan, "GPS spoofing detection using RAIM with INS coupling," in Proceedings of PLANS. IEEE, 2014.

[6] T. Cagatay, K. Samer, and P. Boris, "Impact of Wind Gusts on Detectability of GPS Spoofing Attacks Using RAIM with INS Coupling," in Proceedings of the ION 2015 Pacific PNT Meeting, Honolulu, Hawaii, 2015.

[7] Y. Oshman and M. Koifman, "Robust GPS navigation in the presence of jamming and spoofing," in Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, Texas, 2003.

[8] B. M. Ledvina, W. J. Bencze, B. Galusha, and I. Miller, "An in-line anti-spoofing device for legacy civil GPS receivers," in Proceedings of the 2010 International Technical Meeting of ION, Salt Lake City, UT, 2001.

[9] Y. C. Lee, "Analysis of range and position comparison methods as a means to provide GPS integrity in the user receiver," in Proceedings of the Annual Meeting of ION. Citeseer, Seattle, Washington, 1986.

[10] Y. C. Lee, "Performance of receiver autonomous integrity monitoring (RAIM) in the presence of simultaneous multiple satellite faults," in Proceedings of the 60th Annual Meeting of ION, Albuquerque, NM, 2001.

[11] J. Ni, Y. Zhu, and W. Guo, "An improved RAIM scheme for processing multiple outliers in GNSS," in Proceedings of AINA. IEEE, Niagara Falls, Canada, 2007.

[12] G. Schroth, M. Rippl, A. Ene, J. Blanch, B. Belabbas, T. Walter, P. Enge, and M. Meurer, "Enhancements of the Range Consensus Algorithm (RANCO)," in Proceedings of the 21st International Technical Meeting of ION GNSS, Savannah, GA, 2008.

[13] S. Hewitson and J. Wang, "GNSS Receiver Autonomous Integrity Monitoring (RAIM) for Multiple Outliers," Journal Navigation, vol. 4, pp. 47–57, 2006.

[14] J. Angus, "RAIM with multiple faults," Journal of Navigation, vol. 53, no. 4, pp. 249–257, 2006.

[15] N. L. Knight, J. Wang, and C. Rizos, "Generalized measures of reliability for multiple outliers," Journal of Geodesy, vol. 84, no. 10, pp. 625–635, 2010.

[16] Nikiforov, Igor, Roturier, Benoît, "Advanced RAIM Algorithms: First Results," in Proceedings of the 18th International Technical Meeting of ION GNSS, Long Beach, CA, 2005.

[17] J. Blanch, T. Walter, P. Enge, S. Wallner, F. A. Fernandez, R. Dellago, R. Ioannides, B. Pervan, I. F. Hernandez, B. Belabbas, A. Spletter, M. Rippl, "A proposal for multi-constellation advanced RAIM for vertical guidance," in Proceedings of the 24th International Technical Meeting of ION GNSS, 2011.

[18] B. Juan, T. Walter, P. Enge, Y. Lee, B. Pervan, M. Rippl, A. Spletter, "Advanced RAIM user Algorithm Description: Integrity Support Message Processing, Fault Detection, Exclusion, and Protection Level Calculation," in Proceedings of the 25th International Technical Meeting of ION GNSS, Nashville, TN, 2012.

[19] J. Blanch, T. Walter, P. Enge, Y. Lee, B. Pervan, M. Rippl, A. Spletter, V. Kropp, "Baseline advanced RAIM user algorithm and possible improvements," IEEE Transactions on Aerospace and Electronic Systems, vol. 51, no. 1, pp.713-732, 2015.

[20] Phase II of the GNSS Evolutionary Architecture Study, Feb. 2010. http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/library/documents/media/GEASPhaseII_Final.pdf

[21] P. Papadimitratos and A. Jovanovic, "GNSS-based Positioning: Attacks and Countermeasures," in IEEE MILCOM. IEEE, San Diego, CA, 2008.

[22] H. Wen, P. Y.-R. Huang, J. Dyer, A. Archinal, and J. Fagan, "Countermeasures for GPS Signal Spoofing," in Proceedings of ION GNSS, Long Beach, CA, 2005.

[23] P. Papadimitratos and A. Jovanovic, "Protection and fundamental vulnerability of GNSS," in Proceedings of IEEE IWSSC. IEEE, Toulouse, France, 2008.

[24] B. W. O'Hanlon, M. L. Psiaki, J. A. Bhatti, D. P. Shepard, and T. E. Humphreys, "Real-Time GPS Spoofing Detection via Correlation of Encrypted Signals," Journal of ION, vol. 60, no. 4, pp. 267–278, 2013.

[25] K. Wesson, M. Rothlisberger, and T. Humphreys, "Practical cryptographic civil GPS signal authentication," Journal of ION, vol. 59, no. 3, pp. 177–193, 2012.

[26] J. T. Curran, M. Paonni, and J. Bishop, "Securing the Open-Service: A Candidate Navigation Message Authentication Scheme for Galileo E1 OS," in European Navigation Conference. ENC, Rotterdam, Netherlands, 2014.

[27] 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.

[28] S. Daneshmand, A. Jafarnia-Jahromi, A. Broumandan, and G. Lachapelle, "A low-complexity GPS Anti- Spoofing Method Using a Multi-Antenna Array," in Proceedings of ION GNSS, Nashville, TN, 2012.

[29] J. W. Wisnowski, D. C. Montgomery, and J. R. Simpson, "A comparative analysis of multiple outlier detection procedures in the linear regression model," Journal of Computational Statistics & Data Analysis, vol. 36, no. 3, pp. 351–382, 2001.

[30] D. M. Sebert, D. C. Montgomery, and D. A. Rollier, "A clustering algorithm for identifying multiple outliers in linear regression," Journal of Computational statistics & data analysis, vol. 27, no. 4, pp. 461–484, 1998.

[31] L. Kaufman and P. J. Rousseeuw, Finding groups in data: an introduction to cluster analysis. John Wiley & Sons, 2009, vol. 344.

[32] R. A. Johnson, D. W. Wichern et al., Applied multivariate statistical analysis. Prentice Hall Englewood Cliffs, NJ, 1992, vol. 4.

[33] V. Estivill-Castro, "Why so many clustering algorithms: a position paper," Journal of ACM SIGKDD explorations newsletter, vol. 4, no. 1, pp. 65–75, 2002.

[34] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density- based algorithm for discovering clusters in large spatial databases with noise." in Proceedings of Kdd, 1996.

[35] " GPS/GNSS RINEX data," 2014. [Online]. Available: https://www.unavco.org/data/gps-gnss/ftp/ftp.html

Biographies

Kewei Zhang is a PhD student with the Networked Systems Security group at KTH (Royal Institute of Technology), Sweden.. He earned his MSc in Wireless Systems from KTH and his B.S. in Tele-Communication Engineering from JiLin University, China. His research is concerned with secure localization.

Rashedul Amin Tuhin is expected to earn his MSc in Network Services and Systems from KTH. He earned his BSc in Electrical and Electronic Engineering from the Islamic University of Technology, Bangladesh. He is now a lecturer at the Eastern University, Bangladesh.

Panos Papadimitratos earned his PhD degree from Cornell University, Ithaca, New York, in 2005. He is currently an associate professor in the School of Electrical Engineering at KTH, Stockholm, Sweden, where he leads the Networked Systems Security group.