JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES
www.journalimcms.org

ISSN (Online) : 2454 -7190 Vol.-16, No.-12, December (2021) pp 1-11 ISSN (Print) 0973-8975

IMPLEMENTATION OF NOVEL PGP ALGORITHM FOR ENCRYPTED GPS COMMUNICATION IN SMART CONTAINERS

Mehrunnisa Saleem[1], Sheeraz Ahmed[2], Salman Ahmad[3], Safdar Nawaz Khan Marwat[4], Adnan Khan[5], Muhammad Aadil[6], Said Ul Abrar[7]

[1,3,4,5] University of Engineering & Technology, Peshawar, Pakistan
[2,6] Iqra National University, Peshawar, Pakistan

Email: [1]mehro48@gmail.com

Corresponding Author: **Mehrunnisa Saleem**

https://doi.org/10.26782/jmcms.2021.12.00001

(Received: September 26, 2021; Accepted: November 23, 2021)

Abstract

The ability to check the location of both static and dynamic devices is improving increasingly with each passing day. To track the locations of both static and dynamic machines, Global Positioning System (GPS) is used to exchange the location between the sender and the receiver. However, there are still challenges in the sage and secure transmission and reception of GPS location. The most common challenge is spoofing attack data. This paper proposes the implementation of a Pretty Good Privacy (PGP) encryption algorithm to ensure the safety of GPS packets shared across the communication channels. The GPS location is first encrypted and subsequently sent across a communication channel, which is strong encryption and cannot be decrypted by an unauthorized user.

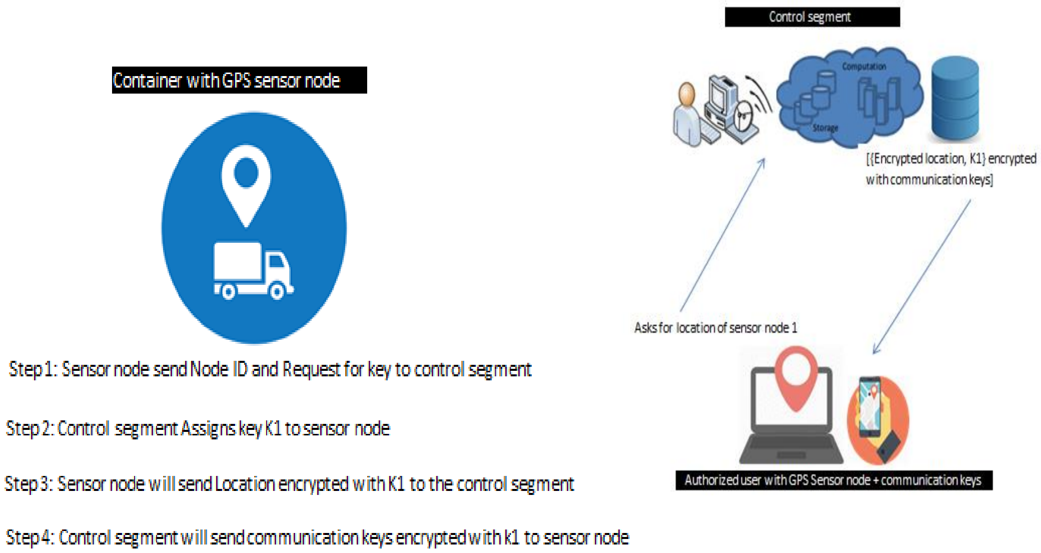
Keywords: Spoofing, Encryption, Decryption, Global Positioning System (GPS)

I. Introduction

In today’s modern world, devices and machines are becoming part of the interconnected network at a much faster rate than ever before. The underlying concept of interconnectivity is sharing of data and information across both wired and wireless communication channels. Data packets across the communication channel without encryption are not secure and the data can be modified or routed to a false location. GPS sharing is too subject to these kinds of attacks. The usage of smart containers which are the basis for today's logistics and supply chain management systems [I],[II], and [III] which manage and transport different packages containing basic daily usage commodities to diligent expensive electronic equipment. Additional the location of public transport systems such as buses and trains has also become important to be monitored as explained in [IV],[V], and [VI]. The majority of devices for sharing their locations utilize GPS devices [VII] which is the largest system used by devices for tracking locations and is estimated to be about used by eight billion devices in 2020[VIII].

GPS devices are vulnerable to spoofing attacks. The contents of the GPS data packets can be changed or removed completely as a result of spoofing attacks to misguide the machines carrying the devices purposedly. In [IX] and [X] the spoofing attacks on a UAV by an unwanted intruder have been discussed in detail. To counter these attacks from the unwanted various techniques are used including the prominent cryptographic method. In [XII],[XIII], and [XIV] and some non cryptographic techniques mentioned in [XV],[XVI],[XVII],[XVIII],[XIX],[XX], and [XXI]. These techniques have been used to prevent the spoofing attacks against the GPS, which are used to prevent spoofing attacks has not been proven effective due to either the raise of false alarms and spoofing notifications or the loopholes which the intruders take advantage of both in hardware and software systems have not packeted are mostly captured by using network data capturing software like Wireshark as mentioned in [XXII],[XXIII], [XXIV], and [XXV] to capture packets. These captured packets are then decoded and decrypted using different brute-forcing techniques to find their encryption key.

The general representation of how the PGP encryption will take place to encrypt the GPS location of the smart containers is shown in Figure 1. The smart container is carrying a machine that has an android operating system installed on it. A GPS device will be embedded with the machine having an android operating system to extract GPS coordinates of the smart container. When the smart container is requested for the GPS location, it extracts the GPS coordinates, and the coordinates are subsequently encrypted using PGP encryption with a key that will be explained in detail in section 4 of this paper. The encrypted GPS location datagram is shared through cloud sharing services. When the encrypted GPS coordinates are received by the user machine, it will decrypt those encrypted data with another key and it is explained in detail in section 4 as to how these unique keys are generated for each user.


Control segment
Computation
Storage
Container with GPS sensor node
[{Encrypted location, K1} encrypted with communication keys]
Asks for location of sensor node 1
Authorized user with GPS Sensor node + communication keys

Step 1: Sensor node send Node ID and Request for key to control segment

Step 2: Control segment Assigns key K1 to sensor node

Step 3: Sensor node will send Location encrypted with K1 to the control segment

Step 4: Control segment will send communication keys encrypted with k1 to sensor node

Figure 1. Showing the general representation of the PGP encryption of the GPS location of the smart containers

II. Implementation of Proposed Algorithm

The following steps have been followed to implement the proposed algorithm for the encryption of GPS location of smart containers using PGP encryption techniques.

i. Research And Analysis

To implement the PGP algorithm, the process of encryption and decryption has to be carried out thoroughly to effectively perform scrambling and unscrambling of data. The inter-conversion of ciphertext and plain text is also a significant part of effectively designing and implementing encryption and decryption of desired data packets. The analysis part is obtained with the help of different simulation techniques to implement a portion of encryption and decryption steps and subsequently integrate all the steps into a working and functional system.

The design part of this proposed algorithm is done using an android application. The android application is designed in flutter and made to run on any device having an android operating system.

For this purpose, the flutter-based designed operating system has to be installed on both the sender and the receiver machine. The sender will be the user requesting the smart container for GPS location. The device embedded into a smart container will be the receiver of the request.

III. Flutter Based Android Application Android Application Interface

To implement a PGP algorithm to encrypt data for secure communication and decrypt it back at the receiving station, a pair of keys are required for both the sender and the receiver. The sender device which is running the application has two keys for PGP encrypted communication. The first key is the private key, which is used to decrypt any received encrypted data from a sender. The second key is the public key which is used to encrypt the data before sending. Both the public and private keys are unique for each user. Any sender can use the publicly available key of the receiver to make encryption with the public key of the receiver. The receiver on receiving the encrypted data will decrypt the encrypted data with its private key. Without a private key, it is next to impossible to decrypt PGP encrypted data. In this application, the user has to be first registered on the android based application using an email ID and password. The application uses the email ID of the user to generate a public key for the registered users once approved by the administrator. Similarly, the application uses the password of the registered user to generate a private key for the registered user and the private key is user-specific.

III. Steps for Implementation of the PGP Algorithm

i. Encryption Process

When a user is authorized to use the application for encrypted. Exchange of GPS location with the smart containers. A pair of keys is assigned to the authorized user. When an authorized user requests for the encrypted GPS location. The smart container will extract its current GPS location using the GPS device. The location is then encrypted with the public key of the requesting user as shown in the block diagram represented by Figure 2. The encrypted file then sends over a communication channel. The advantage of doing this is, that if the encrypted data packets containing the GPS location of the smart container are captured by the intruders for spoofing attack, it is highly unlikely and close to impossible to decrypt it in the absence of the private key of the user.

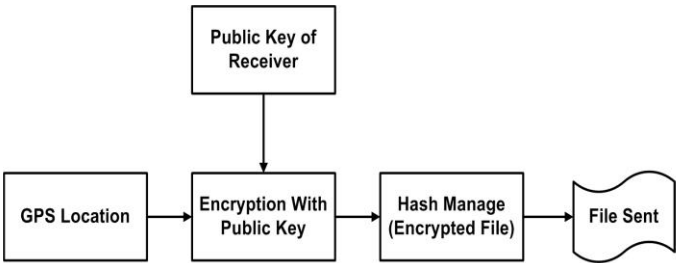


Figure 2. Block diagram of the PGP encryption process

ii. Decryption Process

Upon receiving the encrypted packet by the user from the smart container the packet is first decrypted with the private key of the user, then the GPS location is extracted from the GPS datagram as represented by the block diagram in Figure 3.

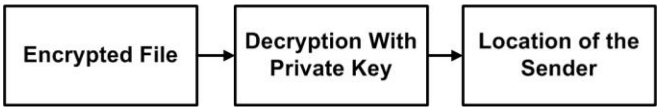


Figure 3. Block diagram of the PGP decryption process

V. PGP Steps for Implementation of Encryption Algorithm

The following steps are used for the implementation of the novel PGP algorithm

i. Authentication

Authentication is the primary and fundamental step in the implementation of the PGP algorithm for encryption and decryption of GPS locations shared by smart containers. This step filters out the unwanted intruders who can potentially harm the system. Authentication is used for safeguarding the data by granting permissions to genuine users and blocking unwanted users. In this android application, the authentication step is implemented by a firebase administrator. The request to use the application for sharing encrypted GPS locations is forwarded to a firebase administrator for approval. The genuine users are granted access to use the application and the application assign a pair of keys including public and private key to genuine users. The private and public keys are assigned to the genuine user for decryption and encryption respectively. In Figure 4 the authentication process is explained. The user has to first sign up to submit the data required in the application, and subsequently, the pending application is forwarded for approval to the firebase administrator.

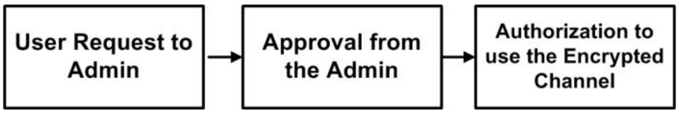


Figure 4. Block diagram representing the process of authentication

ii. Confidentiality

The second step to implement the PGP algorithm is confidentiality. In reaching of data to the intended user is ensured by confidentiality. In this application the confidentiality is implemented by symmetric block encryption is shown in Figure 5. The GPS location is first extracted upon the request of the requesting user, then it is subsequently encrypted with the encryption key to obtain an encrypted GPS location for safeguarding the contents of the GPS location. On the receiver part, once it is received, the decryption process takes place at the receiving end.

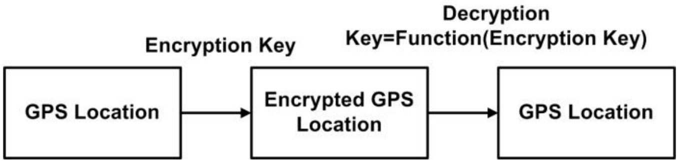


Figure 5. Block diagram for a process of confidentiality

iii. Integrity

Reaching of GPS encrypted data packet to the intended use without changing its encrypted content is integrity conserved. In the implementation of the PGP, algorithm integrity ensures that the contents of the GPS location cannot be changed or modified when sent over a communication channel. The integrity part of the PGP algorithm implementation is carried out using the implementation of digital signature which is a combination of private key encryption and hashing as shown in fig-5. And fig-6 for obtaining a digital signature, hashing process is used to generate hashed GPS location. The private key encryption and hashing are completely obtained when the private key of the user who is sending the request for encrypted GPS location is combined with the hashed GPS coordinates and subsequently digital signature is obtained this way.

V. Results

i. Implementation Of Pgp Encryption and decryption process in android application

The android application designed in flutter can be used by requesting the user machine and the machine installed on the smart container. After signing up on the application the requesting user has to wait for the pending approval. Once the approval is received by the requesting user from the firebase administrator. The flowchart of the program is explained by the Figure given below in Figure 6. As shown in Figure 3, the flow chart is representing the program flow that is running on the application installed on the smart container. The program initially starts, and if a user has requested the firebase administrator for approval. It will wait for the administrator to approve the request of the user. Once the request is approved, the user has to send another request to the smart container program to share its GPS location. Once the request is received, the application extracts the GPS location from the GPS device, and the location is encrypted with the public key of the requesting user, which is received by the program during the request from the user. An encrypted message is created and then subsequently shared with the requesting user.

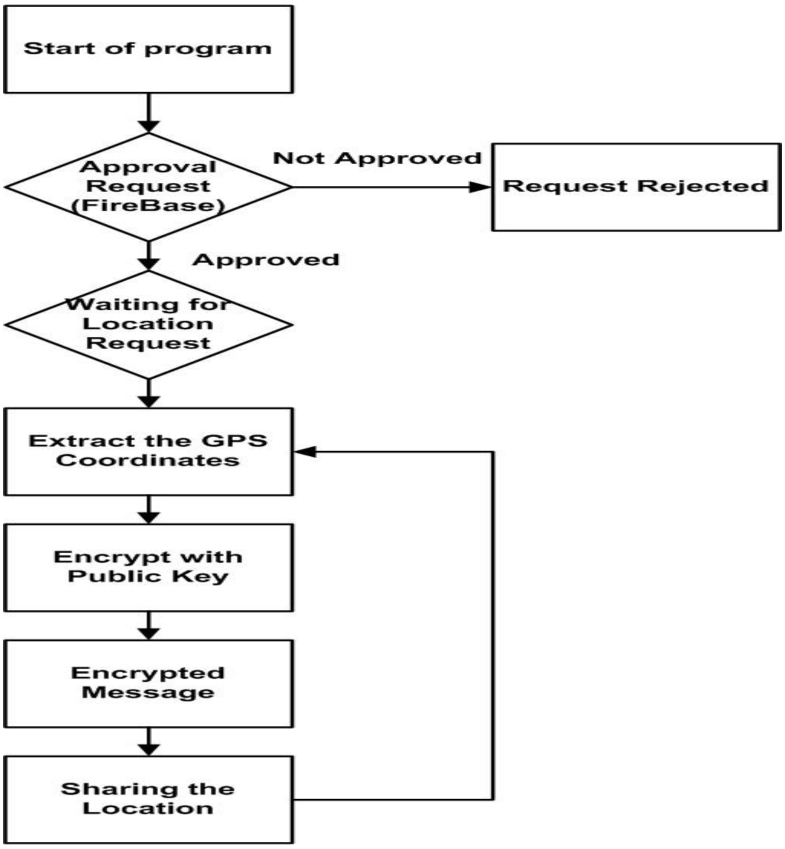


Figure 6. Flowchart for the program running on the smart container

ii. Encryption and decryption process of android application

The android application designed in flutter running on the user machine is used for receiving the encrypted GPS location of the smart containers. This application can receive the encrypted GPS location, and once the location is received, it is subsequently decrypted by the private key of the user as shown in Figure 7. When the program is started, it waits for the encrypted GPS location to be received by the program shared by the GPS device of the smart container. As shown in the flow chart in Figure 6, the received encrypted location is received by the program and subsequently decrypted by the private key of the user, and the location coordinates are then mapped to represent the location of the smart container on the map as shown in Figure 8.

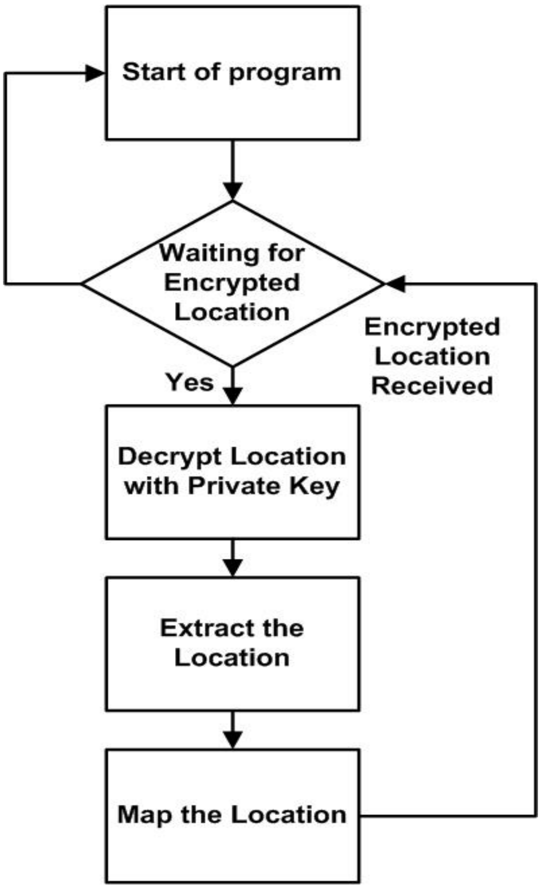

Start of program
Waiting for Encrypted Location
Encrypted Location Received
Yes
Decrypt Location with Private Key
Extract the Location
Map the Location

Figure 7. Flowchart for the program running on the user machine

The coordinates extracted from the GPS location sent by the device in the smart container are then mapped on the GPS location in the android application. Similarly, on the sender side, which is the smart container, also after receiving the request for the GPS location, extracts the coordinates and the encryption option is given as shown in Figure 8 to encrypt the GPS coordinates and send them back to the requesting user. Figure 9 shows the unique public key of every user is available which is used to encrypt the GPS location of the smart container.

VI. Conclusion

In conclusion, the design of the application has improved the GPS location security shared by the smart containers. The location shared by smart containers can't be obtained or changed or modified or deleted by an unwanted intruder. This has become possible due to PGP encryption, and the GPS location can't be obtained by any user until the private key of the user is available

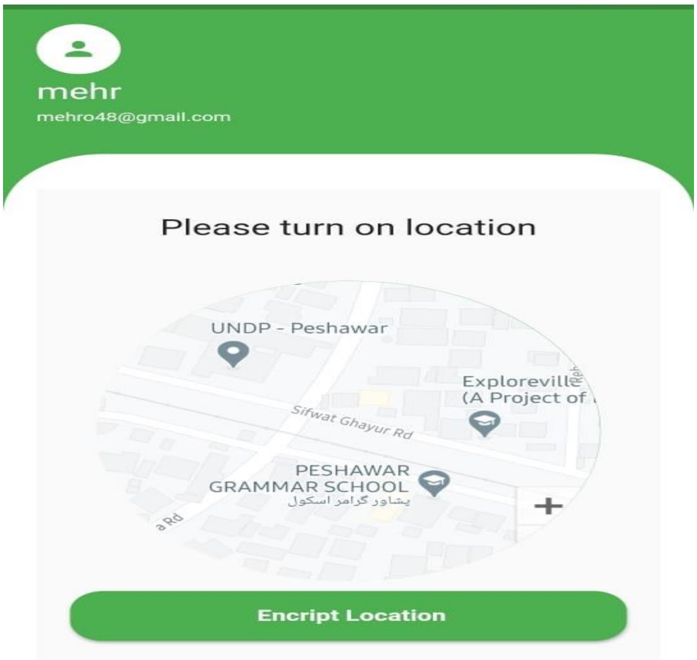
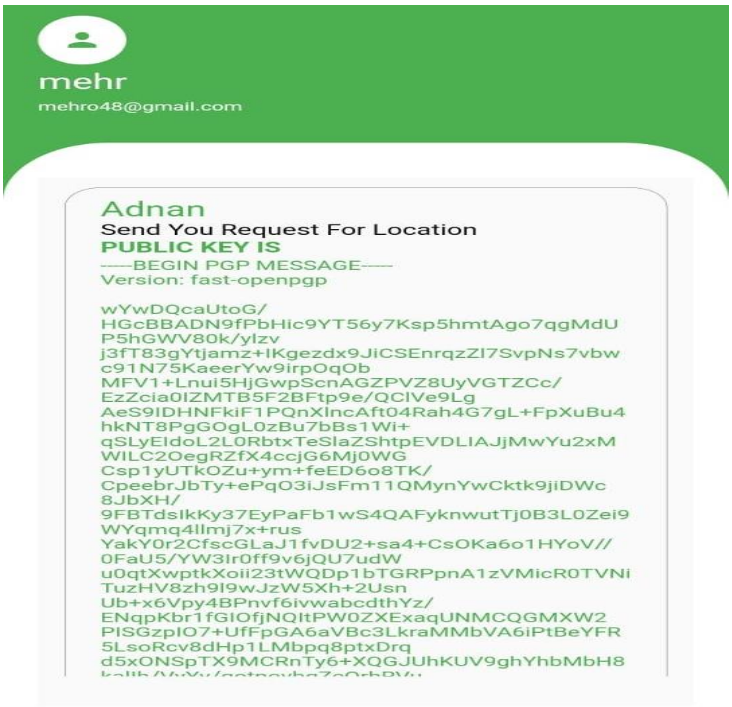


Figure 8. Mapped GPS location of the smart container after decryption.



mehr
mehro48@gmail.com

Adnan
Send You Request For Location
PUBLIC KEY IS

```
-----BEGIN PGP MESSAGE-----
Version: fast-openpgp

wYwDQcaUtoG/
HGcBBADN9fPbHic9YT56y7Ksp5hmtAgo7qgMdU
P5hGWV80k/ylzv
j3fT83gYtjamz+IKgezdx9JiCSEnrqzZl7SvpNs7vbw
c91N75KaeerYw9irpOqOb
MFV1+Lnui5HjGwpScnAGZPVZ8UyVGTZCc/
EzZcia0IZMTB5F2BFtp9e/QClVe9Lg
AeS9IDHNFkiF1PQnXIncAft04Rah4G7gL+FpXuBu4
hkNT8PgGOgL0zBu7bBs1Wi+
qSLyEldoL2L0RbtxTeSlaZShtpEVDLIAJjMwYu2xM
WILC2OegRZfX4ccjG6Mj0WG
Csp1yUTkOZu+ym+feED6o8TK/
CpeebrJbTy+ePqO3iJsFm11QMynYwCktk9jiDWc
8JbXH/
9FBTdsIkKy37EyPaFb1wS4QAFyknwutTj0B3L0Zei9
WYqmq4llmj7x+rus
YakY0r2CfscGLaJ1fvDU2+sa4+CsOKa6o1HYoV//
0FaU5/YW3Ir0ff9v6jQU7udW
u0qtXwptkXoii23tWQDp1bTGRPpnA1zVMicR0TVNi
TuzHV8zh9I9wJzW5Xh+2Usn
Ub+x6Vpy4BPnvf6ivwabcdthYz/
ENqpKbr1fGIOfjNQItPW0ZXExaqUNMCQGMXW2
PISGzpIO7+UfFpGA6aVBc3LkraMMbVA6iPtBeYFR
5LsoRcv8dHp1LMbpq8ptxDrq
d5xONSpTX9MCRnTy6+XQGJUhKUV9ghYhbMbH8
```

Figure 9. The public key of the requesting user to encrypt the GPS location is intended to be sent to the user.

Conflict of Interest:

The authors declare that no conflict of interest to report the present study.

References

I. A. Broumandan, A. Jafarnia-Jahromi, V. Dehghanian, J. Nielsen, and G. Lachapelle, “GNSS spoofing detection in handheld receivers based on signal spatial correlation,” in Proceedings of the IEEE Position Location and Navigation Symposium (PLANS), 2012.

II. A. Dabir and A. Matrawy, "Bottleneck Analysis of Traffic Monitoring using Wireshark," 2007 Innovations in Information Technologies (IIT), 2007, pp. 158-162, doi: 10.1109/IIT.2007.4430446.

III. A. Juels and T. Ristenpart, "Honey encryption: Security beyond the brute-force bound" in Advances in Cryptology-EUROCRYPT 2014, Springer, pp. 293-310, 2014.

IV. A. Juels and T. Ristenpart, "Honey encryption: Security beyond the brute-force bound" in Advances in Cryptology-EUROCRYPT 2014, Springer, pp. 293-310, 2014.

V. A. Ranganathan, H. Olafsd · ottir, and S. Capkun, “Spree: A spoofing · resistant gps receiver,” in Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking. ACM, 2016

VI. Bowen, B.M., Hershkop, S., Keromytis, A.D., Stolfo, S.J.: Baiting Inside Attackers Using Decoy Documents, pp. 51–70 (2009)

VII. D. M. Akos, “Who’s afraid of the spoofer? GPS/GNSS spoofing detection via automatic gain control (AGC),” Navigation, 2012.

VIII. Developing GPS monitoring for the public transport fleet,” http://civitas. eu/measure/developing-gps-monitoring-public-transport-fleet.

IX. E. Schmidt, Z. Ruble, D. Akopian and D. J. Pack, "Software-Defined Radio GNSS Instrumentation for Spoofing Mitigation: A Review and a Case Study," in IEEE Transactions on Instrumentation and Measurement, vol. 68, no. 8, pp. 2768-2784, Aug. 2019, doi: 10.1109/TIM.2018.2869261.

X. F. L. Aryeh, B. K. Alese and O. Olasehinde, "Graphical analysis of captured network packets for detection of suspicious network nodes," 2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), 2020, pp. 1-5, doi: 10.1109/CyberSA49311.2020.9139672.

XI. G. GSA, "Market report issue 3," 2017, https://www.gsa.europa.eu/.

XII. G. Mintsis, S. Basbas, P. Papaioannou, C. Taxiltaris, and I. Tziavos, "Applications of gps technology in the land transportation system," European journal of operational Research, 2004.

XIII. J. Carn, "Smart Container Management: Creating value from real-time container security device data," 2011 IEEE International Conference on Technologies for Homeland Security (HST), 2011, pp. 457-465

XIV. J. Zhang, B. Chen, Y. Zhao, X. Cheng and F. Hu, "Data Security and Privacy-Preserving in Edge Computing Paradigm: Survey and Open Issues," in IEEE Access, vol. 6, pp. 18209-18237, 2018

XV. K. C. Zeng, Y. Shu, S. Liu, Y. Dou, and Y. Yang, "A practical gps location spoofing attack in road navigation scenario," in Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications. ACM, 2017.

XVI. K. K. Songala, S. R. Ammana, H. C. Ramachandruni and D. S. Achanta, "Simplistic Spoofing of GPS Enabled Smartphone," 2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE), 2020, pp. 460-463, doi: 10.1109/WIECON-ECE52138.2020.9397980.

XVII. K. Wesson, D. Shepard, J. 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.

XVIII. K. Wesson, M. Rothlisberger, and T. Humphreys, "Practical cryptographic civil GPS signal authentication," Journal of Navigation, 2012.

XIX. M. A. Poltavtseva, D. P. Zegzhda and E. Y. Pavlenko, "High-performance NIDS Architecture for Enterprise Networking," 2019 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), 2019, pp. 1-3, doi: 10.1109/ Black Sea Com. 2019.8812808.

XX. M. Abadi and B. Warinschi, "Password-based encryption analyzed" in Automata Languages and Programming, Springer, pp. 664-676, 2005.

XXI. M. G. Kuhn, “An asymmetric security mechanism for navigation signals,” in Information Hiding, 2005.

XXII. M. L. Psiaki, S. P. Powell, and B. W. O’Hanlon, “GNSS spoofing detection using high-frequency antenna motion and carrier-phase data,” in Proceedings of the ION GNSS+ Meeting, 2013..

XXIII. P. Misra and P. Enge, Global Positioning System: Signals, Measurements and Performance Second Edition. Lincoln, MA: Ganga-Jamuna Press, 2006.

XXIV. R. Das and G. Tuna, "Packet tracing and analysis of network cameras with Wireshark," 2017 5th International Symposium on Digital Forensic and Security (ISDFS), 2017, pp. 1-6, doi: 10.1109/ISDFS.2017.7916510.

XXV. R. Jedermann, T. Poetsch and W. Lang, ”Smart Sensors for the Intelligent Container,” Smart SysTech 2014; European Conference on Smart Objects, Systems and Technologies, 2014, pp. 1-2

XXVI. S. C. Lo and P. K. Enge, “Authenticating aviation augmentation system broadcasts,” 2010.

XXVII. T. E. Humphreys, “Detection strategy for cryptographic GNSS antispoofing,” IEEE Transactions on Aerospace and Electronic Systems, 2013.

XXVIII. T. Humphreys, “Statement on the vulnerability of civil unmanned aerial vehicles and other systems to civil gps spoofing,” University of Texas at Austin (July 18, 2012), 2012.

XXIX. Tallapalli Chandra Prakash, Srinivas Samala, Kommabatla Mahender. : ‘ MULTICARRIER WAVEFORMS FOR ADVANCED WIRELESS COMMUNICATION’. J. Mech. Cont.& Math. Sci., Vol.-15, No.-7, July (2020) pp 252-259

XXX. US Department of Transportation: In-vehicle Performance Monitoring and Feedback,” https:// www. transportation.gov/ mission/health/ Invehicle -Performance - Monitoring-and-Feedback.

XXXI. W. Yue, Z. Xu and Z. Dapeng, "A High-reliability Network Architecture Based on Parallel Redundancy Protocol," 2019 14th International Conference on Computer Science & Education (ICCSE), 2019, pp. 43-46, doi: 10.1109/ICCSE.2019.8845328.