An ASP-based Approach to Network Security in Urban Air Mobility

Gioacchino Sterlicchio[1,*,†], Francesca Alessandra Lisi[2,*,†]

[1]DMMM, Polytechnic University of Bari, Italy
[2]DIB and CILA, University of Bari Aldo Moro, Italy

Abstract

In this discussion paper, we briefly describe a novel approach to network security, accepted for presentation at ECAI 2024. The approach leverages Answer Set Programming (ASP) for finding contrast sequential patterns that characterize different attacks on the 4G-LTE network in the context of Urban Air Mobility. The experiments show that an ASP-based declarative approach is feasible in this context, and that the implementation of span and gap constraints in the sequence mining phase makes the search for patterns more efficient and effective.

Keywords

Answer Set Programming, Contrast Sequential Pattern Mining, Network Security, Urban Air Mobility

1. Introduction

Urban Air Mobility (UAM) is a new air transportation system for passengers and cargo in urban environments performed by electric aircraft that take-off and land vertically [1]. UAM has a complex ecosystem which makes it vulnerable to cyber-threats putting *network security* of transportation system at risk [2]. *Cellular-connected UAV* [3] is a promising technology to achieve the essential UAM requirements of high-capacity, low-latency and ultra-reliable wireless communications between UAVs and their associated ground entities. 4G long-term evolution (LTE) is an example and can be used together by unmanned aerial system traffic management (UTM) and Automatic Dependent Surveillance Broadcast (ADS-B) ensuring safety and security of the UAV operation [4]. However security threats could lead to a loss of confidentiality (e.g. access and disclosure of restricted information), integrity (e.g. message tampering, service supplier spoofing), or availability (e.g. reliability of GPS) of the data exchanged or stored across the UAM ecosystem [5].

Network security is a field of application for pattern mining algorithms. Buczak *et al.* [6] survey methods of machine learning and data mining that can be successfully applied to intrusion detection. Sequence mining is one of them. Li *et al.* [7] use sequential pattern mining in real time to find out the frequency and sequence features of multi-stage attacks. In [8], the authors construct attack graph from transaction database using sequential pattern mining. Husák *et al.* [9] use sequential pattern mining and rule mining in the analysis of cyber security alert sharing SABU. Whereas sequence mining is widely explored in network security, the *Contrast Sequential Pattern Mining (CSPM)* task [10] has not been addressed so far in this application domain to the best of our knowledge. In this discussion paper, we address the problem of detecting patterns of attacks to 4G-LTE network security in UAM by relying on the CSPM task.

Our approach, described in depth in a paper accepted for presentation at ECAI 2024 [11], leverages the declarative framework of *Answer Set Programming* (ASP) [12], thus positioning in the research stream called *Declarative Pattern Mining* (DPM). The first proposal of an ASP-based approach to sequence mining is described by Guyet *et al.* [13] and compared with a dedicated algorithm. Gebser *et al.* [14]

AIxIA 2024 Discussion Papers - 23rd International Conference of the Italian Association for Artificial Intelligence, Bolzano, Italy, November 25–28, 2024

*Corresponding author.
†These authors contributed equally.
g.sterlicchio@phd.poliba.it (G. Sterlicchio); FrancescaAlessandra.Lisi@uniba.it (F. A. Lisi)
0000-0002-2936-0777 (G. Sterlicchio); 0000-0001-5414-5844 (F. A. Lisi)

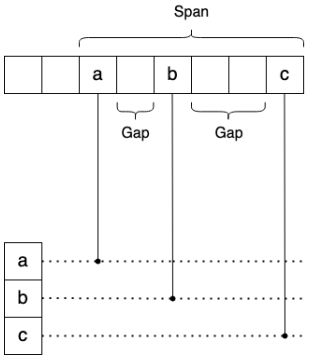

Figure 1: Illustration of the gap and span notions.

use ASP for extracting condensed representations of sequential patterns. Guyet *et al.* [15] introduce ASP encodings for two representations of embeddings (fill-gaps vs skip-gaps) in sequence mining. Lisi and Sterlicchio present the first ASP encoding for the CSPM problem in [16] which we will refer to as *Mining with Answer Set Solving - Contrast Sequential Patterns (MASS-CSP)* hereafter. To address the challenges of DPM in the context of network security, in [11] we have improved the efficiency and the effectiveness of the sequence mining phase in MASS-CSP by adding span and gap constraints.

The paper is organized as follows. In Section 2 and 3 we briefly describe our ASP-based approach to the problem in hand, and report some experimental results obtained on sets of traces for two kinds of attacks. Section 4 concludes the paper with final remarks.

2. Approach

We start with a couple of observations about the limits of *MASS-CSP*. For illustrative purposes, let us consider the pattern $\langle a, b\rangle$ and the sequences $\langle a, b, c\rangle$ and $\langle a, c, c, b\rangle$. First, they do not deal with the number of gaps between one embedding and another. In other words, two consecutive items of a sequential pattern can be n gaps apart within a sequence, in the example 0 and 2 respectively. Secondly, $\langle a, b\rangle$ has support in both sequences but with different span, namely 1 and 3 respectively. Our work develops on the basis of these two observations because in various application domains, patterns that reflect certain characteristics are more informative. In [17] there were defined many types of constraints on patterns and embeddings for sequence mining, among which the ones based on the notion of *gap* and *span* that are detailed below.

The span constraint specifies the minimum/maximum length allowed for a sequential pattern. As illustrated in Figure 1, it is the difference between its last item timestamp that is 8 and its first item timestamp, i.e. 3, and thus $\langle a, b, c\rangle$ has span 5 in that sequence. It requires that the pattern duration should be longer or shorter than a given time period. By setting a span constraint, we can focus on identifying shorter or longer sequences of events based on our specific requirements. The gap constraint controls the minimum/maximum gap allowed between consecutive occurrences of items within a sequence. In Figure 1, the gap between a and b is 1 while 2 between b and c. It specifies how many time units may intervene before an item is observed again. Gap constraints are essential for capturing temporal relationships between events. Setting appropriate gap values helps identify patterns where there might be delays or interruptions between related events but still maintain their significance.

According to [15], we have encoded these constraints as *choices rules* instead of using ASP *denials*, thus implementing them in the generate stage for pruning the search space earlier. The full encoding with other details can be found in [18]. The result is an increase in efficiency and effectiveness of the final output as described in the following section.

3. Evaluation

CSPM can be particularly useful to 4G-LTE for different reasons, e.g., optimizing network performance by analyzing contrast patterns and making informed decisions on network configurations, resource allocations, and traffic management strategies and ensuring quality of services requirements. Since our focus is on security, anomalies or unusual behavior in the network traffic can be detected by mining contrast sequential patterns, thus helping in identifying potential security threats. Contrast sequential patterns are those that characterize normal and attack behaviour given different traces. Our work considers a couple of attacks - namely *authentication failure attack* and the *numb attack* [19] - as a case study and we used the traces made available by [20][1]. Listing 1 shows example patterns for both attacks. More precisely they are the longest contrast sequential patterns found having 30% support across all sequences. They describe the timeline of events that leads to the attack. Interestingly, both attacks share common behavior with the exception of the *authentication_failure* event that occurs always in subsequence $\langle \ldots, \textit{authentication\_request, authetication\_failure, authentication\_request}, \ldots \rangle$ in (a) but not in (b).

Listing 1: Examples of longer attack patterns found with 30% support in (a) Auth_Failure_40 and (b) Numb_Attack_40

```
(a)
<attach_request, authentication_request, authentication_failure,
authentication_request, authentication_response, security_mode_command,
security_mode_complete, attach_accept, attach_complete, detach_request,
detach_accept, attach_request, authentication_request,
authentication_failure, authentication_request, authentication_response,
security_mode_command, security_mode_complete, attach_accept,
attach_complete, detach_request, detach_accept>

(b)
<attach_request, authentication_request, authentication_response,
security_mode_command, security_mode_complete, attach_accept,
attach_complete, detach_request, detach_accept, attach_request,
authentication_request, authentication_response, security_mode_command,
security_mode_complete, attach_accept, attach_complete>
```

The main goal is to show the feasibility of a declarative approach to CSPM in the context of network security. Also, experiments have been designed in order to provide a comparative evaluation between the basic ASP encoding reported in [16] and the ASP encodings proposed that implement the span/gap constraints. We empirically show what are the advantages of adding new constraints on pattern embeddings. Figure 2 makes a comparison between the basic ASP encoding (dotted lines) and the improved one with the span/gap constraints (continuous lines) that for space reasons we only report the authentication failure attack. First, with the gap constraint we have control over the type of pattern we want thanks to the minimum and maximum gap. The pattern output set is considerably reduced, extracting only those actually useful for our purpose with an advantage on time and memory as we act directly in the pattern generation phase, having a smaller ground program than the previous one. Using the span constraint, we are able to reduce the number of patterns and the execution time without memory improvement. The gap constraint is the one that brings the best advantages in terms of overall performance.

4. Conclusion

This discussion paper addresses the problem of detecting attack patterns to the network security in the context of UAM, and focuses on a couple of attacks to 4G LTE, namely the authentication failure and the numb attack. We have suggested that CSPM can be helpful and presented an ASP-based approach to mine contrast sequential patterns from 4G-LTE traces characterizing normal and attack behavior for different types of attacks. The patterns found with our approach may be useful for post-attack analysis in order to understand the steps of the immediate attack and implement defensive mechanisms.

[1] https://github.com/CLC-UIowa/SySLite

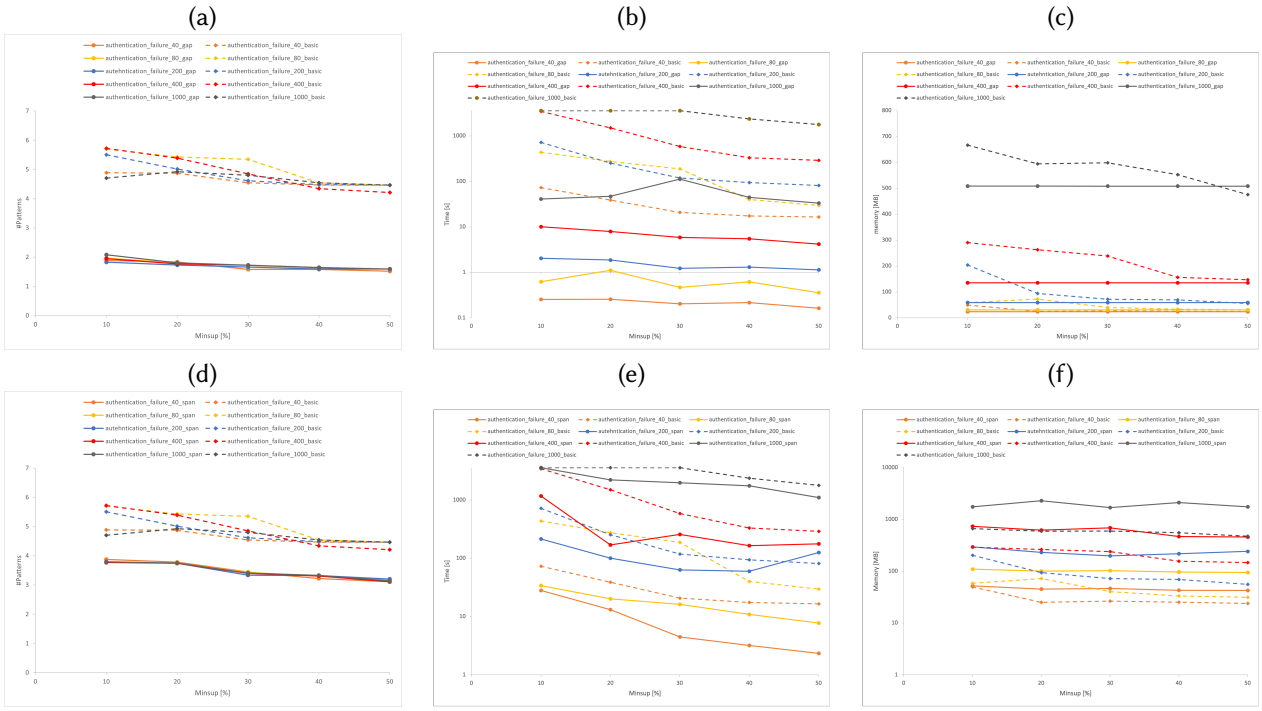

Figure 2: Comparison as regards number of patterns (log scale), execution time, and memory consumption between the basic ASP encoding and the encodings with gap (a-c), span (d-f) constraints on the datasets Auth_Failure with *mingap=0*, *maxgap=0*, *minspan=1*, *maxspan=10 minlen=2*, and *maxlen=6*.

Experiments have shown that applying a declarative approach is feasible. Also, results have highlighted the benefits of adding constraints to embeddings. In particular, the span and gap constraints allow pruning the set of patterns found, thus decreasing the memory consumption and the execution time. Finally, since the only input is the set of execution traces, the approach applies also to other attacks on the same network or even on other networks such as 5G.

Acknowledgments

This work was partially supported by the project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU.

References

[1] H. Pak, L. Asmer, P. Kokus, B. I. Schuchardt, A. End, F. Meller, K. Schweiger, C. Torens, C. Barzantny, D. Becker, J. M. Ernst, F. Jäger, T. Laudien, N. Naeem, A. Papenfuß, J. Pertz, P. Shiva Prakasha, P. Ratei, F. Reimer, P. Sieb, C. Zhu, Can Urban Air Mobility become reality? Opportunities, challenges and selected research results, arXiv e-prints (2023) arXiv:2309.12680. doi:10.48550/arXiv.2309.12680. arXiv:2309.12680.

[2] A. Jordan, K. K. Jaskowska, A. Monsalve, R. Yang, M. Rozenblat, K. Freeman, S. Garcia, Systematic evaluation of cybersecurity risks in the Urban Air Mobility operational environment, in: 2022 Integrated Communication, Navigation and Surveillance Conference (ICNS), IEEE, 2022, pp. 1–15.

[3] Y. Zeng, J. Lyu, R. Zhang, Cellular-Connected UAV: Potential, challenges, and promising technologies, IEEE Wireless Communications 26 (2018) 120–127.

[4] N. Ruseno, C.-Y. Lin, S.-C. Chang, UAS traffic management communications: The legacy of ADS-B, new establishment of remote ID, or leverage of ADS-B-Like systems?, Drones 6 (2022). URL: https://www.mdpi.com/2504-446X/6/3/57. doi:10.3390/drones6030057.

[5] C. Ippolito, K. Krishnakumar, An interface-based cybersecurity subsystem analysis on a small unmanned aerial systems, AIAA SciTech (2019).
[6] A. L. Buczak, E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection, IEEE Communications surveys & tutorials 18 (2015) 1153–1176.
[7] Z. Li, A. Zhang, J. Lei, L. Wang, Real-time correlation of network security alerts, in: IEEE International Conference on e-Business Engineering (ICEBE'07), IEEE, 2007, pp. 73–80.
[8] J. Lei, Z.-t. Li, Using network attack graph to predict the future attacks, in: 2007 Second International Conference on Communications and Networking in China, IEEE, 2007, pp. 403–407.
[9] M. Husák, J. Kašpar, E. Bou-Harb, P. Čeleda, On the sequential pattern and rule mining in the analysis of cyber security alerts, in: Proceedings of the 12th International Conference on Availability, Reliability and Security, ARES '17, Association for Computing Machinery, New York, NY, USA, 2017. URL: https://doi.org/10.1145/3098954.3098981. doi:10.1145/3098954.3098981.
[10] Y. Chen, W. Gan, Y. Wu, P. S. Yu, Contrast pattern mining: A survey, 2022. arXiv:2209.13556.
[11] G. Sterlicchio, F. A. Lisi, Detecting patterns of attacks to network security in urban air mobility with answer set programming, in: U. Endriss, F. S. Melo, K. Bach, A. J. B. Diz, J. M. Alonso-Moral, S. Barro, F. Heintz (Eds.), ECAI 2024 - 27th European Conference on Artificial Intelligence, 19-24 October 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024), volume 392 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2024, pp. 1285–1292. URL: https://doi.org/10.3233/FAIA240626. doi:10.3233/FAIA240626.
[12] V. Lifschitz, Answer sets and the language of answer set programming, AI Magazine 37 (2016) 7–12.
[13] T. Guyet, Y. Moinard, R. Quiniou, Using answer set programming for pattern mining, arXiv preprint arXiv:1409.7777 (2014).
[14] M. Gebser, T. Guyet, R. Quiniou, J. Romero, T. Schaub, Knowledge-based sequence mining with ASP, in: IJCAI 2016-25th International joint conference on artificial intelligence, AAAI, 2016, p. 8.
[15] T. Guyet, Y. Moinard, R. Quiniou, T. Schaub, Efficiency analysis of ASP encodings for sequential pattern mining tasks, in: Advances in Knowledge Discovery and Management, Springer, 2018, pp. 41–81.
[16] F. A. Lisi, G. Sterlicchio, Mining contrast sequential patterns with ASP, in: R. Basili, D. Lembo, C. Limongelli, A. Orlandini (Eds.), AIxIA 2023 - Advances in Artificial Intelligence - XXI-Ind International Conference of the Italian Association for Artificial Intelligence, AIxIA 2023, Rome, Italy, November 6-9, 2023, Proceedings, volume 14318 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 44–57. URL: https://doi.org/10.1007/978-3-031-47546-7_4. doi:10.1007/978-3-031-47546-7_4.
[17] J. Pei, J. Han, W. Wang, Constraint-based sequential pattern mining: the pattern-growth methods, Journal of Intelligent Information Systems 28 (2007) 133–160.
[18] G. Sterlicchio, F. A. Lisi, Detecting Patterns of Attacks to Network Security in Urban Air Mobility with Answer Set Programming, 2024. URL: https://doi.org/10.5281/zenodo.13135192. doi:10.5281/zenodo.13135192.
[19] S. R. Hussain, O. Chowdhury, S. Mehnaz, E. Bertino, LTEInspector: A systematic approach for adversarial testing of 4g LTE, in: 25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018, The Internet Society, 2018. URL: https://www.ndss-symposium.org/wp-content/uploads/2018/02/ndss2018_02A-3_Hussain_paper.pdf.
[20] M. F. Arif, D. Larraz, M. Echeverria, A. Reynolds, O. Chowdhury, C. Tinelli, Syslite: Syntax-guided synthesis of PLTL formulas from finite traces, in: 2020 Formal Methods in Computer Aided Design (FMCAD), 2020, pp. 93–103. doi:10.34727/2020/isbn.978-3-85448-042-6_16.