Declarative Pattern Mining in Network Security

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

This work addresses the problem of detecting patterns of attacks on 4G-LTE network security by relying on the Contrast Sequential Pattern Mining (CSPM) task leveraging the declarative framework of Answer Set Programming (ASP).

Keywords

Answer Set Programming, Declarative Pattern Mining, Contrast Sequential Pattern Mining, Network Security

1. Introduction

Pattern mining is widely applied for different application in network security; see Buczak *et al.* [1] for a survey. Whereas sequence mining is extensively explored in network security, the *Contrast Sequential Pattern Mining (CSPM)* task [2] has not been addressed so far in this application domain to the best of our knowledge. In this extended abstract, we consider the problem of detecting patterns of attacks to 4G-LTE network security relying on the CSPM task.

Our approach, fully described in [3], utilizes the declarative framework of *Answer Set Programming* (ASP) [4], thereby aligning with the research area known as *Declarative Pattern Mining* (DPM) [5, 6, 7]. In particular, Guyet *et al.* [5] describes the first proposal of an ASP-based approach to sequence mining and compares it with a dedicated algorithm. Later, Guyet *et al.* [7] introduce ASP encodings for two representations of embeddings (fill-gaps vs. skip-gaps) in sequence mining.

Lisi and Sterlicchio [8] introduced the initial ASP formulation for the CSPM issue, which we will call *Mining with Answer Set Solving - Contrast Sequential Patterns (MASS-CSP)* from now on. When tackling the DPM challenges related to network security, our work in [3] enhances both the performance and effectiveness of MASS-CSP by incorporating span and gap restrictions during the sequence mining phase. We then perform a comparative evaluation using trace sets from two types of network attack: *authentication failure* and *numb* attacks [9].

The paper is organized as follows. In Section 2 and 3 we briefly describe the *MASS-CSP* approach , and report some evaluation results obtained on sets of traces for the numb attack, respectively. Section 4 concludes the paper with final remarks.

2. The MASS-CSP approach with span/gap constraints

We begin by making a couple of observations regarding the constraints of *MASS-CSP*. To illustrate, consider the pattern $\langle a, b\rangle$ along with the sequences $\langle a, b, c\rangle$ and $\langle a, c, c, b\rangle$. First, the number of gaps between successive embeddings is not addressed. In other words, within a sequence, two neighboring elements of a sequential pattern may be separated by n gaps, which is 0 and 2 in the given example. Secondly, $\langle a, b\rangle$ appears in both sequences but exhibits different spans, specifically 1 and 3, respectively. Our study builds on these insights because in many application areas, patterns exhibiting certain properties are more informative.

Discovery Science - Late Breaking Contributions 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)
© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

CEUR-WS.org/Vol-3928/paper_160.pdf

Many types of constraints on patterns and embeddings for sequence mining are available in the literature [10]. The *span* constraint refers to setting a limit on the length of the sequential patterns being analyzed. This means that when searching for patterns, only those that fall within a specified minimum and maximum length are considered valid. For example, if the minimum length is 3 and the maximum length is 5, patterns shorter than 3 or longer than 5 will not be included in the results. The *gap* constraint involves regulating the gap allowed between consecutive occurrences of items within a sequence. This helps in controlling how close or far apart items in a pattern can appear. For instance, if the minimum gap is 1 and the maximum gap is 3, then each subsequent item in the pattern must appear at least 1 but not more than 3 positions after the previous item. These constraints help in refining the search process to find more relevant or interesting patterns by filtering out those patterns that do not meet the specified criteria.

As noted by [7], we opted to encode these constraints as *choice rules* rather than ASP *denials*, thereby integrating them into the generation phase to reduce the search space proactively. Detailed encoding and additional information are available in [11]. This approach has led to enhanced efficiency and effectiveness of the final output, as discussed in the next section.

3. Experimental results

CSPM can be particularly useful to 4G-LTE. It does this by analyzing sequences of network events to identify patterns that help in making better decisions regarding network configurations, resource allocations, and managing network traffic, all while ensuring that quality of service standards are met. CSPM can detect abnormal or suspicious network activities, which can signify potential security threats. By analyzing these patterns, it is s possible to differentiate between normal and potentially harmful behaviors. Our emphasis is on security, and finding contrast sequential patterns can reveal anomalies or irregularities in network activity, aiding in the identification of possible security risks. These patterns illustrate typical and malicious behavior as reflected in various traces. Our work considers the *authentication failure* attack and the *numb* attack as a case study for which we used the traces made available by [12][1]. As an illustration, Listing 1 shows an example of contrast sequential pattern for the *numb attack*. It is the longest pattern found having 30% support across all sequences, and describes the timeline of events that leads to the attack.

Listing 1: Example of pattern found with 30% support in Numb_Attack_40.

```
<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 primary aim of the evaluation is to demonstrate the practicality of using a declarative approach for CSPM within network security. Additionally, the experiments are structured to offer a comparative assessment between the original MASS-CSP documented in [8] and its enhanced version incorporating the span/gap constraints. In [3], we provide empirical evidence demonstrating the benefits of introducing additional constraints on pattern embeddings. Figure 1 compares the standard MASS-CSP (represented by dotted lines) with the enhanced MASS-CSP (depicted by continuous lines) in the context of the numb attack.

By setting minimum and maximum gap constraints, the process refines and controls the types of patterns generated, ensuring that only relevant patterns are produced. This refinement reduces computational output and enhances efficiency by saving time during the pattern generation phase. However, in terms of memory usage, it is not much variation as the values remain similar or slightly increase. Overall, the gap constraint is highlighted for its performance improvements, particularly in minimizing extraneous patterns and expediting the process without adversely affecting memory usage significantly. By implementing the span constraint, we can decrease the number of patterns and reduce

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

execution time. However, this approach does not lead to any improvements in memory usage and may actually require more storage space. In contrast, using the gap constraint offers the greatest benefits, significantly enhancing overall performance.

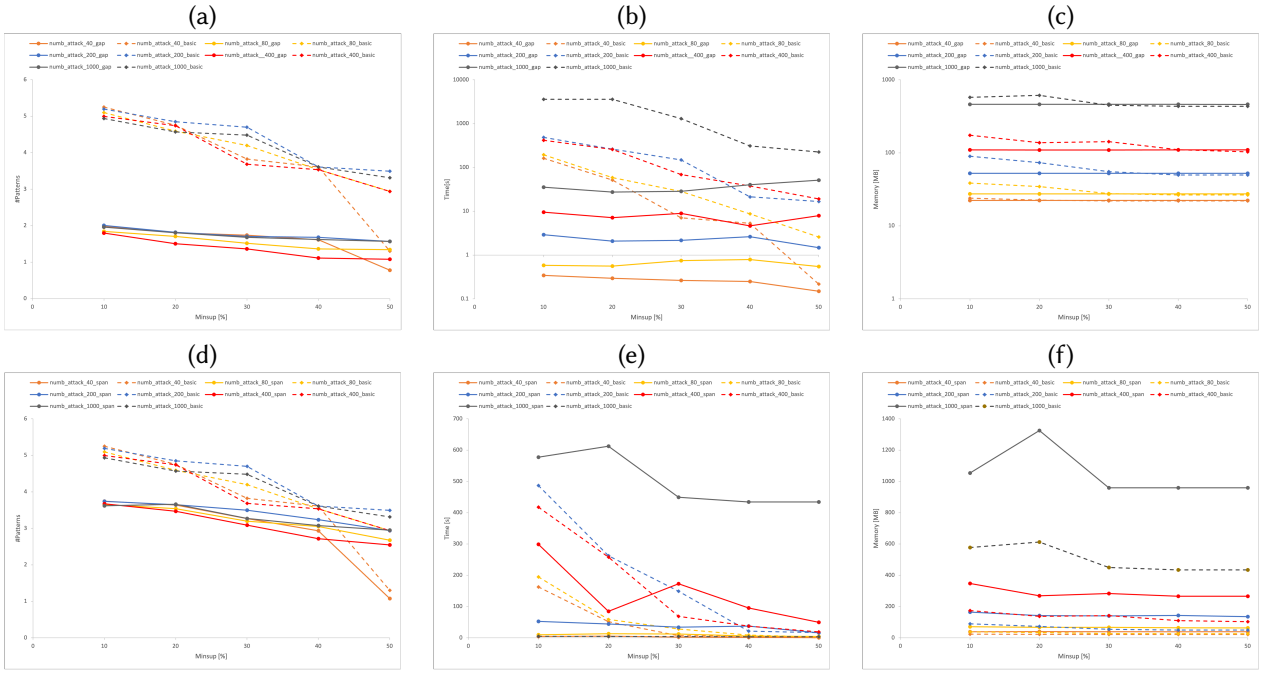

Figure 1: Comparison as regards number of patterns (log scale), execution time, and memory consumption between the basic ASP encoding and the encodings with gap (a-c), span (d-f) constraints on the datasets Numb_Attack with *mingap=0*, *maxgap=0*, *minspan=1*, *maxspan=10*, *minlen=2*, and *maxlen=6*.

4. Conclusion

This research investigates the detection of attack patterns of the 4G LTE network security, focusing on attacks like authentication failure and the numb attack. We leverage Contrast Sequential Pattern Mining (CSPM) implemented using Answer Set Programming (ASP) to analyze 4G LTE network traffic behavior. This method allows us to identify sequential patterns that differentiate between normal and attack-related network behavior. The resulting patterns are valuable for understanding attack progression and creating effective countermeasures.

The research demonstrates that using a declarative method for detecting network security attacks is possible. By incorporating specific constraints — namely span and gap constraints — the system becomes more efficient by reducing the number of patterns it needs to process. This leads to lower memory usage and faster execution times. Additionally, because the approach is based solely on analyzing execution traces, it is highly adaptable and can be applied to various types of attacks across different network systems, including advanced technologies like 5G. This versatility ensures that the method remains relevant and effective as network technologies evolve.

Acknowledgments

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

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] 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.

[2] Y. Chen, W. Gan, Y. Wu, P. S. Yu, Contrast pattern mining: A survey, 2022. arXiv:2209.13556.

[3] 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.

[4] V. Lifschitz, Answer sets and the language of answer set programming, AI Magazine 37 (2016) 7–12.

[5] T. Guyet, Y. Moinard, R. Quiniou, Using answer set programming for pattern mining, arXiv preprint arXiv:1409.7777 (2014).

[6] 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.

[7] 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.

[8] 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 - XXIInd 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. doi:10.1007/978-3-031-47546-7_4.

[9] 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.

[10] J. Pei, J. Han, W. Wang, Constraint-based sequential pattern mining: the pattern-growth methods, Journal of Intelligent Information Systems 28 (2007) 133–160.

[11] 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.

[12] 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.