A design space exploration approach to jointly optimize security and schedulability in TSP systems

Ill-ham Atchadam, Frank Singhoff, Hai Nam Tran, Laurent Lemarchand
Univ. of Brest, UMR 6285, Lab-STICC, F-29200 Brest, France
Email: {ill-ham.atchadam, frank.singhoff, hai-nam.tran, laurent.lemarchand}@univ-brest.fr

I. Introduction

Modern aircraft integrate more and more functions to provide more services to users. To master such complexity, the integrated modular avionics (IMA) architecture [1] proposes to share the computing resources between the software functions of the aircraft. In that case, a fault occurring in a function can easily affect others without isolation mechanisms. IMA architecture solves this through time and space partitioning (TSP). Space isolation is obtained by memory protection between partitions while time isolation is enforced by off-line partitioned scheduling.

[***Problem statement***] Avionic functions implemented as real-time tasks in TSP systems have stringent constraints on safety, security, and schedulability. A TSP system is a set of applications made of tasks. For fault propagation limitation, tasks are assigned to partitions that are software units defined to ensure temporal and spatial isolation.

The design of TSP systems requires deciding how to assign tasks to partitions. The number of possible assignments increases exponentially with the number of tasks. Furthermore, changing the tasks to partitions assignments has an impact on the schedulability of the system.

TSP systems may host applications provided by different stakeholders with a significant level of legacy, which increases the probability of corrupted or malicious software deployment. Data exchanged between tasks can be intercepted during application communications which result in either confidentiality violations by disclosure of sensitive information or integrity violations by data alteration. Data confidentiality can be achieved by encryption algorithms [2] and data integrity can be achieved by hashing algorithms [3]. However, ensuring data confidentiality and integrity incurs a significant computation overhead on banalized hardware. This overhead affects the system schedulability by leading some tasks to miss their deadlines.

[***Contribution***] In this work, we address the conflict between schedulability and security by proposing a Design Space Exploration (DSE) approach for unicore TSP systems based on the Pareto Archived Evolutionary Strategy (PAES) meta-heuristic [4] that provides trade-offs for multi-objective optimization problems (MOOP). Furthermore, we experiment the extensibility of our DSE for multicore execution platforms.

[***Related work***] Many works have addressed the security of real-time systems. Several papers proposed security optimization ([5], [6]) while others focused on scheduling optimization ([7], [8]). In [9], the authors consider a joint optimization by proposing trade-offs. DSE approaches for real-time systems have also been proposed by many researchers such as [10] and [11]. However, as far as we know, no work has proposed a DSE that addresses security and schedulability optimization for TSP systems.

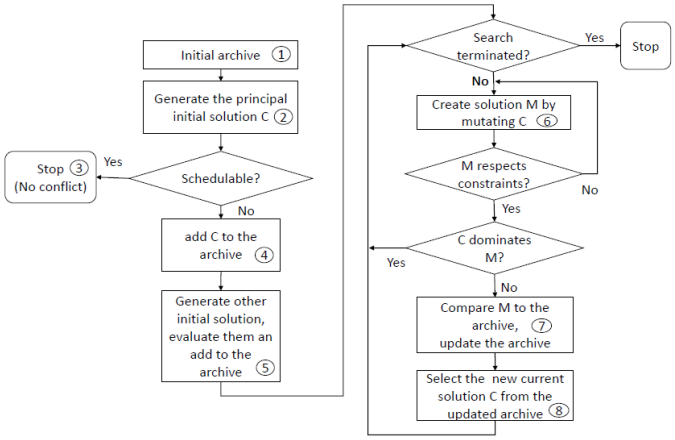

Fig. 1: PAES process

II. Design space exploration

We propose an approach to explore the search space of TSP while investigating tasks and partitions assignment and communications security in order to address the conflict between schedulability and security. The objective is to minimize two conflicting objective functions, deadlines misses, and communications vulnerabilities. Deadlines misses of low criticality tasks and a subset of unsecured communications are tolerable. Communications can be secured by applying different techniques. We adopt PAES to explore the search space defined by the constraints and the objectives and find an approximate set of optimal solutions in a suitable time for large-scale problems that are very time-consuming with an exact method (i.e. exhaustive research).

Figure 1 presents our adaptation of the PAES to schedulability and security optimization problem. For each step of the process, a specification has to be given depending on the MOOP addressed. PAES is an iterative process where the DSE is performed by generating a candidate solution based on a current solution. It starts with an empty archive (step 1) and an initial solution (step 2) defined based on the MOOP problem and the addressed systems.

The initial solutions (steps 4 and 5) correspond to various tasks to partitions assignments, e.g solving or not all security vulnerabilities in the system, placing all the initial tasks in the same partition, or balancing them on multiple partitions and running on a single or multiple cores.

At the first iteration, the current solution is the initial solution which consists of the entry point of the DSE process. During the DSE, at each iteration, a mutation operator is applied to the current

solution to generate a candidate solution (step 6).

We propose three mutation algorithms: *task-grain*, *app-grain*, and *mix-grain*. The first one changes randomly locations of tasks to partition. The second one restricts locations of whole applications, i.e. all tasks of an application are assigned on the same partition. The last algorithm operates firstly at application level and then refines in a second phase the solutions by mutating individual tasks. The mutation also changes communications. We randomly choose a low communication and switch its status from secured to unsecured and vice-versa, with also a randomized security implementation, depending notably on the locations (same partition or not) of the communicating tasks.

Each candidate solution is evaluated for the two objective functions and goes through feasibility tests in order to determine if it respects the schedulability and security constraints. Missed deadlines are computed by scheduling simulation [12], and security vulnerabilities correspond to both integrity and confidentiality rules violations according to Bell-La Padula [13] and Biba [14] principles.

At each iteration, each feasible candidate solution is compared to the current solution and the other solutions in the archive (step 7) based on the Pareto dominance principle [15] considering the values of the objective functions. The archive is updated in order to keep only non-dominated solutions. Then a solution is selected to become the current solution of the next iteration (step 8).

III. Results

Experiments are conducted to evaluate our proposed approach. First, the results show that for applications with a low processor utilization and small messages such as control-command applications, there is no need for DSE because securing them fully will not affect schedulability. Second, we show the effectiveness of our approach in providing trade-offs for models where full security leads to missed deadlines. As expected, our results confirm that processor utilization, exchanged data size, number of partitions, and communications overheads are key parameters that affect the trade-off between security and schedulability. A comparison with the exhaustive research for a case study confirms the effectiveness of our approach in providing good solutions in a reasonable amount of time.

Third, we compare the three mutation algorithms *task-grain*, *app-grain*, and *mix-grain*. Results show that *mix-grain* can propose interesting solutions impossible to be generated with *app-grain* because they are out of its design space (e.g. solutions with tasks of the same application split into different partitions). We see that the design space of *task-grain* is too large to converge towards the best solutions. The *mix-grain* algorithm can be seen as a solution to the problem of convergence of *task-grain* since both have the same search space and granularity level.

Finally, the exhaustive research also confirms the effectiveness of *mix-grain* because by proposing solutions with tasks of the same application assigned to different partitions not reachable by *app-grain*.

IV. Conclusion

In this article, we propose a DSE approach based on the PAES to trade-offs on the task partitioning according to security and schedulability objective functions for TSP systems. We show the effectiveness of our approach with different experimentations. We also show the extensibility of our approach by applying it to another context: conflict between security and schedulability in TSP systems on multicore execution platforms while considering safety constraints (i.e. active redundancy). Considering multicore execution platforms in TSP implies not only tasks to partitions assignment but also tasks to cores assignment. In our experiments, we only consider interconnection overheads due to multicore execution platforms. For future work, we intend to consider the other overhead including main memory and cache on these platforms.

References

[1] C. B. Watkins and R. Walter, "Transitioning from federated avionics architectures to integrated modular avionics," in *2007 IEEE/AIAA 26th Digital Avionics Systems Conference*. IEEE, 2007, pp. 2–A.

[2] J. Thakur and N. Kumar, "Des, aes and blowfish: Symmetric key cryptography algorithms simulation based performance analysis," *International journal of emerging technology and advanced engineering*, vol. 1, no. 2, pp. 6–12, 2011.

[3] Y. Zheng, J. Pieprzyk, and J. Seberry, "Havala one-way hashing algorithm with variable length of output," in *International workshop on the theory and application of cryptographic techniques*. Springer, 1992, pp. 81–104.

[4] J. Knowles and D. Corne, "The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, vol. 1. IEEE, 1999, pp. 98–105.

[5] V. Lesi, I. Jovanov, and M. Pajic, "Security-aware scheduling of embedded control tasks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 5s, pp. 1–21, 2017.

[6] T. Xie and X. Qin, "Scheduling security-critical real-time applications on clusters," *IEEE transactions on computers*, vol. 55, no. 7, pp. 864–879, 2006.

[7] Q. N. Ahmed and S. V. Vrbsky, "Maintaining security in firm real-time database systems," in *Proceedings 14th Annual Computer Security Applications Conference (Cat. No. 98EX217)*. IEEE, 1998, pp. 83–90.

[8] Q. Xue, Y. Zhu, Y. Wang, K. Mao, H. Wu, M. Li, Y. Mao, and J. Hou, "A scheduling scheme of task allocation in real time multiple-partition embedded avionic," in *2017 IEEE International Conference on Smart Cloud (SmartCloud)*. IEEE, 2017, pp. 41–46.

[9] S. H. Son, R. Mukkamala, and R. David, "Integrating security and real-time requirements using covert channel capacity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 6, pp. 865–879, 2000.

[10] R. Bouaziz, L. Lemarchand, F. Singhoff, B. Zalila, and M. Jmaiel, "Multi-objective design exploration approach for ravenscar real-time systems," *Real-Time Systems*, vol. 54, no. 2, pp. 424–483, 2018.

[11] M. Hasan, S. Mohan, R. Pellizzoni, and R. B. Bobba, "A design-space exploration for allocating security tasks in multicore real-time systems," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2018, pp. 225–230.

[12] F. Singhoff, J. Legrand, L. Nana, and L. Marcé, "Cheddar: a flexible real time scheduling framework," in *ACM SIGAda Ada Letters*, vol. 24, no. 4. ACM, 2004, pp. 1–8.

[13] D. E. Bell and L. J. La Padula, "Secure computer system: Unified exposition and multics interpretation," MITRE CORP BEDFORD MA, Tech. Rep., 1976.

[14] K. J. Biba, "Integrity considerations for secure computer systems," MITRE CORP BEDFORD MA, Tech. Rep., 1977.

[15] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based pareto optimality for many-objective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 269–285, 2013.