Secure Multi-Party Computation with Service Contract Automata

by Davide Basile (ISTI-CNR)

By combining research from model-based software engineering, dependable computing, and formal methods, it is possible to create a contract-based design methodology to enforce security accountability and reputation of distributed digital entities provided by potentially mutually distrusted organisations.

Our society is increasingly dependent on heterogeneous digital infrastructures, for example in the healthcare, financial and transport domains. These infrastructures are examples of systems of systems, i.e., they are realised through the composition of several sub-systems provided by potentially mutually distrusted or competing organisations. An example is the ERTMS/ETCS Level 3, a new railway signalling system where virtual positioning is replacing legacy physical systems, and the geolocation sub-system is provided by a third party (e.g., European GNSS service). Moreover, emerging computing paradigms (e.g., fog, mobile-edge or cloud computing, to mention a few) rely on components discovered and accessed over the internet. The composed behaviour needs to be validated to guarantee overall security, as well as safety and interoperability requirements.

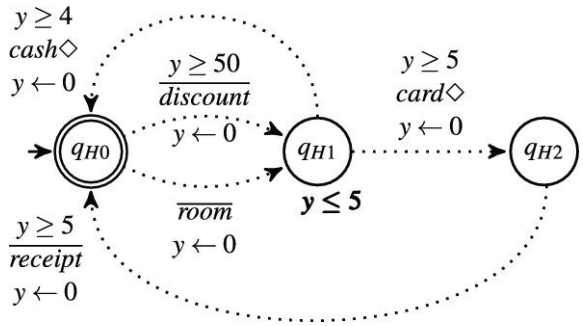

Figure 1: An example of a real-time service contract automaton.

In these emerging multi-party paradigms, no assumption shall be made about third-party systems, which are accessed as a black box. Thus, standard monolithic verification techniques used for validating digital entities cannot be applied to ensure the overall security of these digital infrastructures. Indeed, novel formal verification techniques must cope with the unwanted scenario where a verified system does not comply with its expected behaviour (called contract), either unintentionally or maliciously, or when the necessary security measures are not in place. For cyber-physical systems, we also cannot make assumptions about the open physical environment in which these systems are operating, whose behaviour (e.g., delays of radio communications, geo-positioning uncertainty) could be tampered with by attackers to drive an unprepared system to unsafe configurations to carry out the attack.

Service contracts [1] have been introduced in the literature as a methodology for designing systems where the requirements and obligations of each party are rigorously specified and rendered as formal specifications, e.g., automata (see Figure 1). The security threats are thus considered from the early design phases of a system. This can reduce costs by detecting design flaws as early as possible, for example, unsecure assumptions on other components or the environment (e.g., stochastic distribution on delays or positioning errors). Contracts are digital entities that must be composable to predicate over their aggregate multi-party behaviour. Firstly, it is necessary to check whether the requirements of each contract are satisfied in the composition by some other contract. Traces leading to violation of the requirements must be pruned to obtain a composition where all involved parties adhere to the shared behaviour. Starting from a raw composition of contracts, this contract agreement can be synthesised automatically [2]. The contract agreement is then proposed to each involved party for validation or further formal verification, before starting their interactions, to ensure the correctness and security of their composed behaviour. Due to the stochastic, physical nature of phenomena involved in a cyber-physical infrastructure, a challenge is to investigate novel formalisms and verification techniques for specifying and verifying contracts expressing both the discrete and continuous aspects under analysis, as well as the stochastic physical phenomena involved.

During the computation, the agreement is realised through coordination of distributed software entities, to allow them to fulfil both their specified requirements and declared obligations. The coordination is generally achieved using a choreographic or orchestrated approach [3], and the requirements for realising a choreography are more stringent because each party must be able to fulfil its requirements and duties independently of the other entities involved in the overall computation. This amounts to project the global agreement to each local component, and if the proper conditions are satisfied, using the local contract agreement instead of the global one to check that each component fulfils its agreed behaviour. On the other hand, an orchestration drives the involved parties to realise their agreed computation, at the cost of extra coordination interactions. In the orchestrated approach, the orchestration is responsible for guaranteeing that the contract agreement is realisable. Thus, the orchestration needs to be a trusted software component.

The key aspect is that the multi-party computation can be monitored at runtime by exploiting the contract agreement each party has signed beforehand,

to detect possible contract breaches in case obligations are not fulfilled, providing log information for post-mortem analyses. Indeed, it is assumed that each party can fulfil its contract and is responsible for violations. In this scenario, it becomes possible to identify the organisations liable for providing services breaching the contract they have agreed upon. This is at the basis of a methodology for formally specifying systems able to guarantee the necessary security accountability and reputation mechanisms for digital organisations.

An open-source API available at [L1] has been produced by the author for developing contract-based applications using a model-driven state-based design approach for software applications built around their specified contract. An open-source graphical application for designing contract specifications, composing them and synthesising a coordination policy in agreement is available at [L2], which has been developed using the API in [L1].

This research activity is partially funded by the PRIN 2017 project "IT MaTTerS: Methods and Tools for Trustworthy Smart systems", funded by the Italian Ministry of Education, University and Research, where the synthesis of run-time monitors is addressed, and by the 4SECURail project "FORmal Methods and CSIRT for the RAILway sector", targeting the construction of railway infrastructures whose sub-systems are provided by different railway companies. 4SECURail received funding from the Shift2Rail Joint Undertaking (JU) under the European Union's Horizon 2020 research and innovation programme under grant agreement No. 881775.

Links:
[L1] https://kwz.me/h6E
[L2] https://kwz.me/h6K

References:
[1] D. Basile: "Specification and Verification of Contract-Based Applications" (Ph.D. thesis, Department of Computer Science, University of Pisa), 2016. hhttps://kwz.me/h7i
[2] D. Basile, et al.: "Controller synthesis of service contracts with variability", Science of Computer Programming, 187; 2020. https://kwz.me/h7h
[3] D. Basile, M. H. ter Beek, R. Pugliese: "Synthesis of Orchestrations and Choreographies: Bridging the Gap between Supervisory Control and Coordination of Services", Logical Methods in Computer Science, 16, 2020. https://lmcs.episciences.org/6527

Please contact:
Davide Basile, ISTI-CNR Pisa, Italy
davide.basile@isti.cnr.it

Towards Privacy-Preserving Sharing of Cyber Threat Intelligence for Effective Response and Recovery

by Lasse Nitz, Mehdi Akbari Gurabi, Avikarsha Mandal and Benjamin Heitmann (Fraunhofer FIT)

Many European organisations suffer from a lack of sufficient resources to provide satisfactory and timely response and recovery (R&R) actions when targeted by cyber-attacks. R&R capabilities can be significantly improved through sharing of information related to incident detection and handling. In this context, privacy-preserving technologies can enable data sharing, while protecting privacy- and security-critical information. The technologies to achieve this are being developed and evaluated in the SAPPAN project.

The computer security incident response team (CSIRT) plays a crucial role in an organisation's digital infrastructure. One of the responsibilities of a CSIRT is to detect, investigate, and mitigate potentially security-critical incidents. To help with the vast number of potential threats, many CSIRTs rely on partly automated systems, especially for the detection of incidents. Since the quality of these detection systems has a direct impact on the manual workload of incident handlers, who have to investigate the detected incidents, the false-positive rate of the incident detection should be as low as possible. The same applies to the false-negative rate, as every undetected incident might pose a serious security risk to an organisation. There is hence a need for high-quality detection system components, which detect incidents reliably without unnecessarily increasing the investigative workload of human operators. But since considerable effort is required to create such high-quality components, it is unfeasible for many small and medium-sized enterprises (SMEs) to create them on their own.

This problem could be overcome by the sharing of cyber-threat intelligence that helps detect, assess and handle incidents, for example as trained classifiers or cybersecurity playbooks. An abstract overview of a sharing system is shown in Figure 1. For security providers, this could constitute a meaningful way of extending their services, and for academic organisations it would allow research results to be made usable in practice. The main problem in sharing resources, however, is that they are usually based on privacy- and security-critical data, so it is vital that no sensitive information can be extracted from the shared resources.

While anonymisation and sanitisation solutions exist for various kinds of data within the cybersecurity domain (e.g., for IP addresses), other kinds of data – for example, uniform resource locators (URLs) – have not received the same level of attention. While URLs have been used in research, e.g., for the identification of phishing websites [1],