Towards a Security, Privacy, Dependability, Interoperability Framework for the Internet of Things

Othonas Soultatos*†, George Spanoudakis*§, Kostantinos Fysarakis§,
Ioannis Askoxylakis†, George Alexandris†, Andreas Miaoudakis† and Nikolaos E. Petroulakis†*
*School of Informatics, City University London, London, UK
†Institute of Computer Science, Foundation for Research and Technology Hellas, Iraklio, Greece
§Sphynx Technology Solutions AG, Zug, Switzerland
{othonas.soultatos, g.e.spanoudakis}@city.ac.uk, fysarakis@sphynx.ch,{asko,alexandris,miaoudak,npetro}@ics.forth.gr

***Abstract*—A popular application of ambient intelligence systems constitutes of assisting living services on smart buildings. As intelligence is imported in embedded equipment, the system becomes able to provide smart services (e.g. control lights, air-conditioning, provide energy management services etc.). IoT is the main enabler of such environments. However, the interconnection of these cyber-physical systems and the processing of personal data raise serious security and privacy issues. In this paper we present a framework that can guarantee Security, Privacy, Dependability and Interoperability (SPDI) in IoT. Taking advantage of the underlying IoT deployment, the proposed framework not only implements the requested smart functionality but also provide modelling and administration that can guarantee those SPDI properties. Moreover, we provide an application example of the framework in a smart building scenario.**

***Index Terms*—security, privacy, dependability, interoperability, IoT**

I. Introduction

Internet of things (IoT) applications and their enabling platforms are often vulnerable to security attacks. IoT applications operating and context conditions changes can compromise their security [1]. They can also generate, make use of, and inter-relate massive personal data in ways that can potentially breach legal and privacy requirements [1]. In the context of the ever-evolving IoT threat landscape [2], preserving security and privacy properties remains a particularly challenging problem, due to the difficulty in:

- analysing vulnerabilities in the complex end-to-end compositions of heterogeneous smart objects
- selecting appropriate controls (e.g. different schemes for ID and key management, different encryption mechanisms), for smart objects with heterogeneous resources/constraints [3], [4]
- preserving end-to-end security and privacy under dynamic changes in IoT applications and security incidents

The above challenges give rise to significant complexities, and relate to the implementation and deployment of IoT applications with embedded intelligence across all layers, as indicated in Fig.1. To address those challenges we propose a pattern-driven framework, built upon existing IoT platforms,

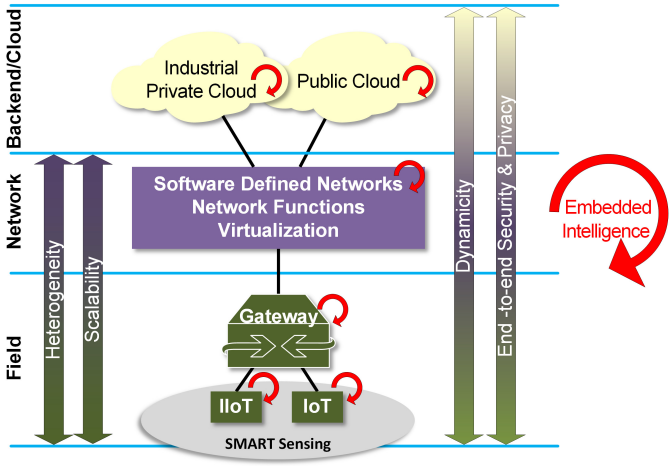

Fig. 1. IoT challenges - Embedded cross-layer intelligence

able to enable and guarantee secure and dependable actuation and semi-autonomic behaviour in both IoT and Industrial IoT (IIoT) applications. The approach followed in the proposed work is inspired by the work carried out in SEMIoTICS, a horizon 2020 project.

The remainder of this paper is organized as follows. Section II presents an overview of the related work and how the proposed framework extends their current state of the art. Section III describes the vision and ambition in deploying a fully re-configurable framework. Section IV explains the system level design of the framework in an smart building use case. Finally, section V summarises the paper.

II. Background and Related Work

A. Patterns and SPDI

Patterns are reusable solutions to common problems and building blocks to architectures. In our approach, patterns encode proven dependencies between Security, Privacy, Dependability and Interoperability (SPDI) properties of individual smart objects and corresponding properties of orchestrations

(composition) involving them. The encoding of such dependencies enables:

- the verification that a smart object orchestration satisfies certain SPDI properties
- the generation (and adaptation) of orchestrations in ways that are guaranteed to satisfy required SPDI properties.

Our approach to patterns is inspired from similar pattern-based approaches used in service-oriented systems [5], [6] , cyber physical systems [7], [9], and networks [8], [10]–[12], [15].

An emerging approach to securing software systems, known as ”security-by-design”, aims to guarantee system-wide security properties by virtue of the design of software systems. Sectet [13], for instance, supports the design of orchestrations of software system components and services as UML message charts and converts them into workflows after wrapping services within components enforcing security properties, known as Policy Enforcement Points. PWSSec [14], uses security architectural patterns to convert security requirements into specifications of service-oriented architectures that make use of external security services providing required security functionalities. A key capability required in security-by-design is the ability to verify the desired security properties as part of the design process. Verification has often been model-based [16]–[18]. Model based approaches verify the satisfiability of the properties based on model checking [19], [20]. In these approaches, software component and service compositions are modelled using formal languages and the required security properties are expressed as properties on the model [21]. Other approaches focus on software service workflows using business process modelling languages (e.g., Sec-MoSC [22]) Pino et al [23] use secure service orchestration (SSO) patterns to support the design of service workflows with required security properties.Their approach uses pattern-based analysis to verify security properties, avoiding full model checking that is computationally expensive and non-scalable to large system. Other model-based approaches (e.g., [23], [24]) support the transformation of security requirements to code. In [25], the authors present a practical and easy to use methodology to measure security, privacy and dependability.

III. Vision and Ambition

In this work, we are presenting a dynamically configurable and evolvable framework to enable: (a) the integration of heterogeneous smart objects that are available through heterogeneous IoT platforms into IoT applications in a manner that is scalable, secure, privacy preserving and dependable; (b) the provision of multi-layer intelligence capabilities enabling semi-autonomic smart object behaviour and evolution; and (c) the runtime management and adaptation of these objects and the IoT applications that they form to preserve security, privacy, and dependability.

Figure 2 shows our initial vision of the initial logical architecture of the SEMIoTICS framework, which motivated the work presented herein, as defined at the start of the project, and how it relates to smart objects, IoT applications, and existing IoT platforms, and how this architecture maps

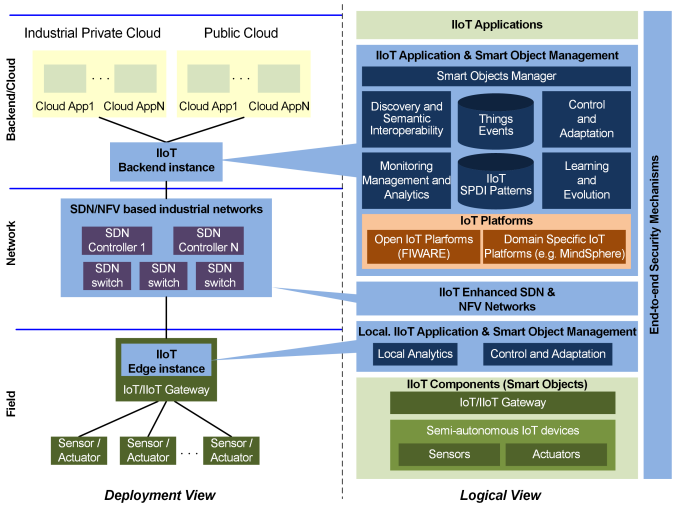

Fig. 2. Framework Architecture

onto a generic deployment infrastructure consisting of private and public clouds, networks, and field devices. Within the figure, blue boxes show components of the framework that are to be developed; white boxes indicate components of IoT applications managed by the framework. The key role of our framework in the IIoT/IoT implementation stack is to support the secure, dependable and privacy-preserving connectivity and interoperability of IoT applications and smart objects used by them, and the management, monitoring and adaptation of these applications, objects and their connectivity. Our framework will support cross-layer intelligent dynamic adaptation, including heterogeneous smart objects, networks and clouds. To address the complexity and scalability needs within horizontal and vertical domains, we will develop and integrate smart programmable networking and semantic interoperability mechanisms leveraged by a Software Defined Networking (SDN) deployment [26], [27].

The key element enabling our approach is the use of architectural SPDI patterns. These patterns define generic ways of composing (i.e., establishing the connectivity between) and configuring the heterogeneous smart objects and software components that may exist at all layers of the IoT applications implementation stack, including: sensors and actuators; smart devices; software components at the network, cloud, IoT enabling platforms and/or other middleware layer; as well as software components at the IoT application layer. To do so, patterns specify abstract and generic smart object interaction and orchestration protocols, enhanced (if necessary) by transformations to ensure the semantic compatibility of data.

Furthermore (and more importantly), the smart object interaction and orchestration protocols encoded by the patterns must have proven ability (i.e., an ability proven through formal verification or demonstrated through testing and/or operational evidence) to achieve not only a semantically viable interoperability between the smart objects that they compose but also specific security, privacy, dependability and interoperability

(SPDI) properties, which may be required of compositions. The compositions defined by patterns are both vertical and horizontal, i.e., they can involve smart objects at the same (horizontal) or different layers (vertical) layer of the IoT implementation stack.

As an example of a pattern that guarantees "data integrity" i.e., absence of unauthorized modifications of data consider the integrity preserving cascade composition pattern discussed in [28]. According to this pattern in a sequential composition of processes P_1,...,P_n where the input data of P_i are meant to be the output data of P_{i-1}, and the communication between P_{i-1} to P_i (i=2...n) is based on an orchestrator O which facilitates data transfers from P_{i-1} to P_i, overall data integrity is preserved if data integrity is preserved within each P_i, within O and across all communications from P_is to O and vice versa. The integrity cascade composition pattern applies both to horizontal compositions (e.g., in software services workflows as in [28]) and vertical composition (e.g., in transfer of data in invocation of operations of IoT enabling middleware) [29].

Another (more complex) example of a pattern fitting our vision is the Synchronously Controlled Distribution Line (SCDL) pattern discussed in [7]. SCDL guarantees that a distributed asynchronous sensor system installed upon a physical pipeline (e.g., a pipeline of an electricity distribution network) will operate in virtual synchrony and provide a guaranteed density of readings (i.e., a bounded minimum number of readings per distant and per time unit). The application of the SCDL pattern is proven to guarantee the consumption of readings at the end of the reading interval where they fit, make them available in a synchronous manner, filter out illegitimate readings and produce readings of the required density for the pipeline. In SCDL pattern, these properties are guaranteed even in the presence of missing or corrupted raw data, as long as there is a minimal number of legitimate sensor readings.

Examples of additional patterns have been given in [5] and [12], [15]. These include patterns for confidentiality in service orchestrations and patterns for availability in Software Defined Networks, respectively.

Inspired by these earlier works, we will define patterns specifying:

- Composition structures for integrating smart objects and components of IoT enabling platforms in a manner that guarantees SPDI properties.
- The end-to-end SPDI properties that the compositions expressed by the pattern preserve.
- The component level SPDI properties that the types of smart objects and/or components orchestrated by the pattern, must satisfy in order to preserve the end-to-end SPD properties.
- Additional conditions that need to be satisfied for guaranteeing end-to-end SPDI properties. These may, for example, include configuration conditions that need to be satisfied by the IoT platforms and the networks providing the connectivity between them, for guaranteeing the end-to-end availability properties of IoT application (composition).
- Monitoring checks that must be monitored at runtime in order to verify that any assumptions about the individual smart objects and components that are orchestrated by a pattern or other operational conditions, which are critical for the preservation of the end-to-end SPDI properties of the pattern, hold at runtime.
- Adaptation actions that may be undertaken to adapt IoT applications, which realise the composition structure of the pattern, at runtime. Such actions may, for example, include the replacement of individual smart objects within a composition; the adaptation of the process realizing the composition; the modification of the configuration of the network services used to connect the smart objects of the composition and/or the deployment platforms upon which these objects run. Adaptation actions are specified along with guard conditions determining when they can be executed (guards are monitored and adaptation is triggered when they are satisfied).

It should be noted that the final framework solution will also include a generic engine (i.e., Smart Object Manager in Figure 2) supporting the execution of patterns at runtime to realize the overall process of monitoring, forming, adapting and managing smart object orchestrations in IoT applications.

The system can be configured at runtime in order to retain the desirable SPDI goals [30]. This operation is important, especially in cases of cyber-attacks. Intrusion detection systems [31] discover ongoing attempts to infiltrate the smart building infrastructure and alert the involved components. Then, the SPDI controller (placed in the SDN controller) is able to alter the system architecture automatically, based on pre-defined strategies, and enhance protection. Also, a remote management service enables the system administrator to change the system manually and update the reaction plans based on the latest security guidelines (i.e. made by CERTs).

Thus, the users would consume safely the deployed services. In case of attack, their security will be retained and the personal data will be safeguarded as the defined SPDI patterns guarantee the principles of "E2E protection".

IV. Smart Home Scenario

The application of the proposed SPDI patterns in a smart building scenario is depicted in Figure 3. In this scenario, an SDN controller maintains in the Knowledge Base (KB) the core SPDI properties for every individual component of the underlying subsystem. Then, the controller estimates the status of the entire system (Smart objects, Services, Applications etc.). This way the controller can permit/block composition activities, or even can change configurations at runtime in order to comply with the designed SPDI properties. As aforementioned, the administrator can also perform the same functionality through a remote management service and update the controller's reactive strategies.

Figure 4 describes the sequence of events for an incident response when a cyber-attack is performed (e.g. unauthorized access attempt). Once a threat is identified, an Intrusion Detection System (IDS) raises an internal alarm to inform

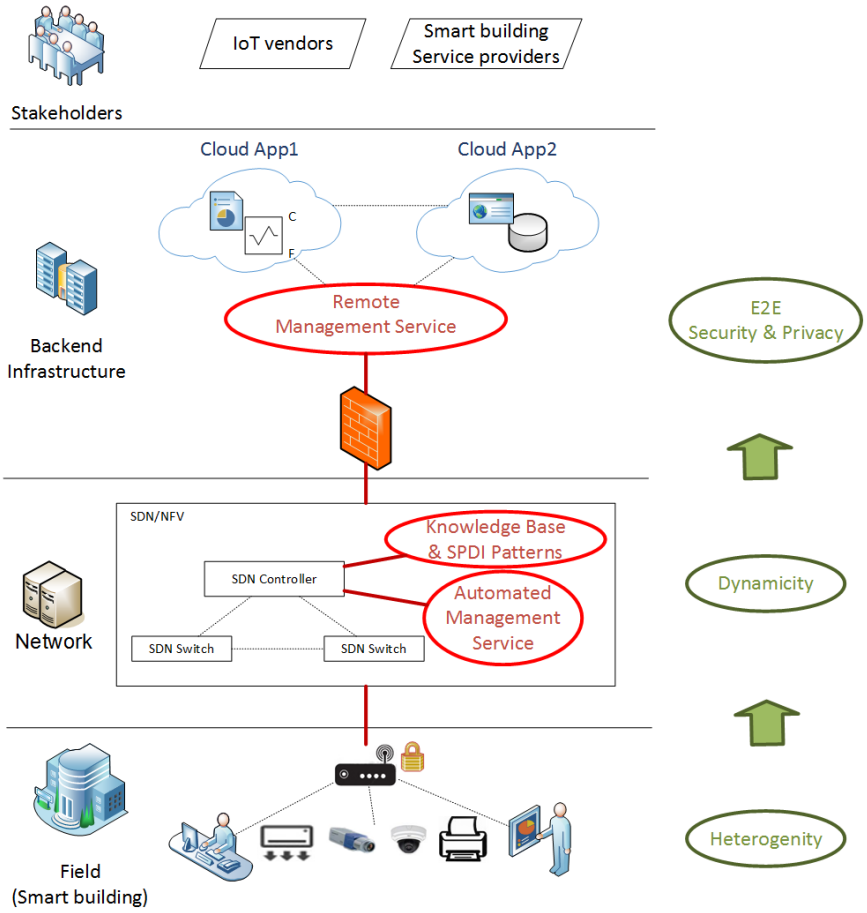

Fig. 3. Smart Building Use Case

the involved system components, which is then taken into account during the response phase, and counters the attack automatically. The administrator is notified for the incident and, after data analysis, and reports the event to the responsible Computer Emergency Report Team (CERT). The CERT performs thorough analysis and disseminates the new guidelines for mitigating such attacks to the relevant action team. The team updates the response policy and the administrator deploys the new SPDI patterns to the SDN controller, which adapts the system to the new state. The reactive plan should constrain the effects of the attack or even counter it, while preserving the desired SPDI goals. The produced data of the incident response phase (e.g. triggering events, taken actions) are maintained in the knowledge base for feedback and reference point of future similar events.

A. *Technical aspects*

The technical aspects of the use case includes the actors, information exchanged between them and also some assumptions and preconditions that apply to the presented smart building showcase. The main actors of the framework and their responsibilities are the following:

- SDN Controller: monitors the network traffic and manages the underlying components based on SPDI goals. The controller enforces strategies for ensuring specific SPDI patterns and achieves E2E properties. In case of attack, the controller automatically configures the system at runtime to preserve protection and an adequate SPDI level.
- Administrators: monitor and manage the system, detect anomalies, and control incidents. When a security incident occurs, administrators can forward it to the Computer Emergency Report Team (CERT) which is capable in following specific strategies and multi-level security response.
- CERTs: cooperate with Internet service providers, ICT vendors, and government agencies, and provide several services to their member and the public, including: i) collection and dissemination of information related to computer security, ii) raising of information security awareness, iii) in depth analysis of security incidents, iv) event handling and response, v) collaboration with other CERTs.
- Action Teams: handle security incidents that are reported by a CERT. Among others, they also: i) install mechanisms for filtering the network traffic, ii) detect intrusion actions in the system, iii) perform actions for protecting

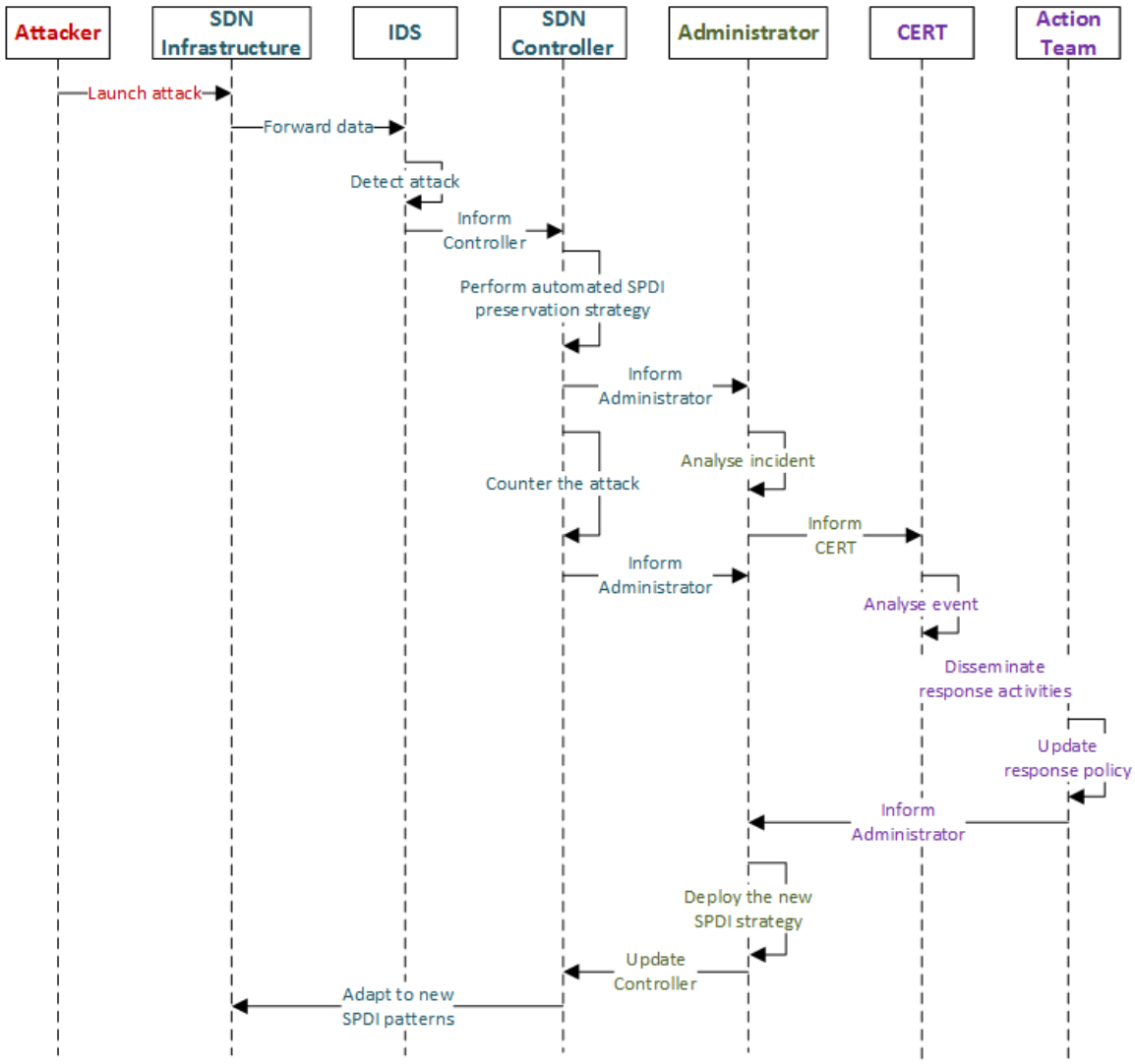

Fig. 4. Incidence Report Sequence

the system threatened/affected by malicious activity, iv) patch the system, v) restore affected systems, vi) offer solutions from related advisories and alerts vii) deploy other response policies.

- End Users

For the demonstrated use case, some assumptions are made. First of all, all SPDI properties of every framework's component are evaluated during the design phase. Moreover an incident strategy describing how information is passed and the action to be taken during an attack should be deployed. Also the response of the CERTs during an attack not only contains the procedure of activities to be taken but also defines the new SPDI perspectives maintananse policy. Furthermore, it is assumed that the framework is able to monitor and collect cyber data from the components such as security logs and system and file integrity information. Also the reported data from the sensory equipment such as critical alerts must be available for analysis from the framework in order to ensure SPDI levels.

Data exchanged between actors should be secured. The IDS produces an automated event monitoring capture report and the information transfer methods between IDS and SDN Controller are determined by the capabilities of the device that run the IDS. They may include: VPN, HTTPS, FTPS, FTP over SSH, IPSec, and SFTP. Furthermore as presented at [32] all sensors can send their traffic through secure channels. As aforementioned controller automatically processes the data and classify the incident based on pre-established mechanisms and the Knowledge Base that securely maintains the previously detected events. Then the administrator is notified respectively and logins to the system to access the stored information. The administrator logs in the CERTs incident handling system through an HTTPS session and the event is recorded and forwarded for further process by the CERT experts.

V. Conclusion

In this work, we propose a framework that adopts a pattern-based approach to guarantee SPDI properties across horizontal and vertical compositional structures of IoT applications. SPDI patterns in this approach set necessary and sufficient conditions not only for composing different components within IoT applications in ways that guarantee SPDI properties, but also for ensuring that IoT applications will use the framework in ways that guarantee such properties. The used SPDI patterns extend existing work on patterns by covering in an integrated manner not only security, but also dependability, privacy, and interoperability properties focusing on IoT systems and applications. A smart building application of the proposed framework is also presented.

Acknowledgement

This work has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No. 780315 (SEMIoTICS), as well as the Marie Skodowska-Curie Actions (MSCA) Research and Innovation Staff Exchange (RISE), H2020-MSCA-RISE-2017, under grant agreements No. 777855 (CE-IoT) and No. 778229 (Ideal Cities).

References

[1] Kert M. et al., State of the Art of Secure ICT Landscape, NIS Platform WG 3, V2, April 2015

[2] ENISA Threat Landscape Report 2016, https://www.enisa.europa.eu/publications/enisa-threat-landscape-report-2016

[3] K. Rantos, A. Papanikolaou, K. Fysarakis, Ch. Manifavas. "Secure policy-based management solutions in heterogeneous embedded systems networks", 2012 International Conference on Telecommunications and Multimedia (TEMU), pp 227-232, 2012, DOI:10.1109/TEMU.2012.6294723

[4] K. Fysarakis, I. Papaefstathiou, Ch. Manifavas, K. Rantos, O. Sultatos, "Policy-based access control for DPWS-enabled ubiquitous devices" Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA), 2014, DOI:10.1109/ETFA.2014.7005233

[5] Pino L., et al. "Pattern Based Design and Verification of Secure Service Compositions." IEEE Transactions on Services Computing (2017).

[6] Pino L., Spanoudakis G., Fuchs A., Gurgens S., "Discovering Secure Service Compositions". 4th International Conference on Cloud Computing and Services Sciences (CLOSER 2014), Barcelona, Spain, April 2014

[7] A. Maa, E. Damiani, S. Grguens, G. Spanoudakis. "Extensions to Pattern Formats for Cyber Physical Systems." Proceedings of the 31st Conference on Pattern Languages of Programs (PLoP14. Monticello, IL, USA. Sept. 2014.

[8] K. Fysarakis et al. "RtVMF: A Secure Real-Time Vehicle Management Framework," in IEEE Pervasive Computing, vol. 15, no. 1, pp. 22-30, Jan.-Mar. 2016. doi: 10.1109/MPRV.2016.15

[9] N. Petroulakis, G. Spanoudakis, I. Askoxylakis, A. Miaoudakis and A. Traganitis, "A pattern-based approach for designing reliable cyber-physical systems", IEEE Global Communications Conference (GLOBECOM), 2015

[10] G. Hatzivasilis et al. "AmbISPDM: Managing Embedded Systems in Ambient Environment and Disaster Mitigation Planning," Applied Intelligence, Springer, vol. 48, issue 6, pp. 1623-1643, 2018. https://doi.org/10.1007/s10489-017-1030-0

[11] G. Hatzivasilis, I. Papaefstathiou and C. Manifavas, "SCOTRES: Secure Routing for IoT and CPS," in IEEE Internet of Things Journal, vol. 4, no. 6, pp. 2129-2141, Dec. 2017. doi: 10.1109/JIOT.2017.2752801

[12] N. Petroulakis, G. Spanoudakis , and I. Askoxylakis "Patterns for the design of secure and dependable software defined networks." Computer Networks 109 (2016): 39-49

[13] Gartner, 7 Technologies Underpin the Hype Cycle for the Internet of Things, 2016, The challenges of creating, implementing and preparing for the IoT, Nov 2016, http://www.gartner.com/smarterwithgartner/7-technologies-underpin-the-hype-cycle-for-the-internet-of-things-2016/

[14] C. A. Gutierrez, E. Fernandez-Medina and M. Piattini, "PWSSEC: Secure Web Services-based Systems Development Process," in IEEE Latin America Transactions, vol. 4, no. 2, pp. 115-122, April 2006. doi: 10.1109/TLA.2006.1642459

[15] N. Petroulakis, G. Spanoudakis, I. Askoxylakis, "Fault Tolerance Using an SDN Pattern Framework", 2017 IEEE Global Communications Conference (GLOBECOM), 2017

[16] Bartoletti M, et al. "Semantics-based design for secure web services." Software Engineering, IEEE Trans. on, 2008.

[17] Deubler M., et al. "Sound development of secure service-based systems." In Proc. of the 2nd Int. Conf. on Service oriented computing. ACM, 2004.

[18] Geor G., et al. "Verification and trade-off analysis of security properties in UML system models." IEEE Trans. on Software Engineering, 36(3): 338-356, 2010.

[19] Siveroni I., Zisman A., Spanoudakis G.: "A UML-Based Static Verification Framework for Security, Requirements", Engineering Journal, 15(1): 95-118, 2010.

[20] Rossi, S.; Model checking adaptive multilevel service compositions; International Workshop of Formal Aspects of Component Software 2010.

[21] Dong, J., et al, Automated verification of security pattern compositions. Inf. Softw. Technol., vol. 52, no. 3, 2010.

[22] Andre R. Souza, et al. Incorporating Security Requirements into Service Composition: From Modelling to Execution, in ICSOC-ServiceWave '09, 2009.

[23] Pino L., Mahbub K., Spanoudakis G., "Designing Secure Service Workflows in BPEL, 12th International Con oriented architectures." In Proc. of IEEE Int. Conf. on Web Services 2010.

[24] Sguran M, Hbert C, and Frankova G. "Secure workflow development from early requirements analysis." In Proc. of IEEE 6th European Conf. on Web Services. 2008.

[25] G. Hatzivasilis, I. Papaefstathiou and C. Manifavas, "Software Security, Privacy, and Dependability: Metrics and Measurement", in IEEE Software, vol. 33, no. 4, pp. 46-54, July-Aug. 2016. doi: 10.1109/MS.2016.61

[26] K. Ramantas, E. Kartsakli,M. Irazabal, A. Antonopoulos, C. Verikoukis : "Implementation of an SDN-Enabled 5G Experimental Platform for Core and Radio Access Network Support." Auer M., Tsiatsos T. (eds) Interactive Mobile Communication Technologies and Learning. IMCL 2017. Advances in Intelligent Systems and Computing, vol 725. Springer, Cham

[27] K. Fysarakis, O. Soultatos,Ch. Manifavas, I. Papaefstathiou, I. Askoxylakis, "XSACdCross-domain resource sharing & access control for smart environments" Future Generation Computer Systems, vol. 80, 2018, DOI:10.1016/j.future.2016.05.023

[28] Pino L., Spanoudakis G., Fuchs A., Gurgens S., Discovering Secure Service Compositions, 4th International Conference on Cloud Computing and Services Sciences (CLOSER 2014), Barcelona, Spain, April 2014.

[29] G. Hatzivasilis, "Password-Hashing Status," Cryptography, MDPI Open Access Journal, vol. 1, issue 2, number 10, 2017, DOI:10.3390/cryptography1020010

[30] K. Fysarakis,G. Hatzivasilis. I. Askoxylakis, Ch. Manifavas,"RT-SPDM: Real-Time Security, Privacy and Dependability Management of Heterogeneous Systems" ,Tryfonas T., Askoxylakis I. (eds) Human Aspects of Information Security, Privacy, and Trust, HAS 2015, Lecture Notes in Computer Science, vol 9190, 2015, DOI:10.1007/978-3-319-20376-85_5

[31] A. Antonopoulos, C. Verikoukis, "Misbehavior detection in the Internet of Things: A network-coding-aware statistical approach" In Proc. 2016 IEEE 14th International Conference on Industrial Informatics (INDIN), 2016

[32] G. Hatzivasilis, E. Gasparis, A. Theodoridis, C. Manifavas, "ULCL – An Ultra-lightweight Cryptographic Library for Embedded Systems," PECCS, 7-9 January, 2014, Lisbon, Portugal, pp. 11-18, DOI:10.5220/0004900602470254