7

Large Scale IoT Security Testing, Benchmarking and Certification

Abbas Ahmad[1,5], Gianmarco Baldini[2], Philippe Cousin[1], Sara N. Matheu[3], Antonio Skarmeta[3], Elizabeta Fourneret[4] and Bruno Legeard[4,5]

[1]Easy Global Market, France
[2]JRC, Italy
[3]University of Murcia, Spain
[4]Smartesting Solutions & Services, France
[5]FEMTO ST/Université de Bourgogne Franche-Comté, France

Abstract

The Internet of Things (IoT) is defined by its connectivity between people, objects and complex systems. This is as vast as it sounds spanning all industries, enterprises, and consumers. The massive scale of recent Distributed Denial of Service (DDoS) attacks (October 2016) on DYN's servers that brought down many popular online services in the US, gives us just a glimpse of what is possible when attackers are able to leverage up to 100,000 unsecured IoT devices as malicious endpoints. Thus, ensuring security is a key challenge. In order to thoroughly test the internet of things, traditional testing methods, where the System Under Test (SUT) tested pre-production, is not an option. Due to their heterogeneous communication protocol, complex architecture and insecure usage context, IoTs must be tested in their real use case environment: service based and large-scale deployments.

This article describes the challenges for IoT security testing and presents a Model Based Testing approach solution, which can be used to support and EU security certification framework at European level for IoT products.

7.1 Introduction

The Internet-of-Things (IoT) is rapidly heading for large scale meaning that all mechanisms and features for the future IoT need to be especially designed and duly tested/certified for large-scale conditions. Also, Security, Privacy and Trust are critical conditions for the massive deployment of IoT systems and related technologies. Suitable duly tested solutions are then needed to cope with security, privacy and safety in the large scale IoT. Interestingly, world-class European research on IoT Security & Trust exists in European companies (especially SME) and academia where even there are available technologies that were proven to work adequately in the lab and/or small-scale pilots. More, unique experimental IoT facilities exist in the EU FIRE initiative that make possible large-scale experimentally-driven research but that are not well equipped to support IoT Security & Trust experiments.

But notably, Europe is a leader in IoT Security & Trust testing solutions (e.g. RASEN toolbox, ETSI Security TC, etc.) that can be extended to large-scale testing environments and be integrated in FIRE IoT testbeds for supporting experimentations. The ARMOUR project aims at providing duly tested, benchmarked and certified Security & Trust technological solutions for large-scale IoT using upgraded FIRE large-scale IoT/Cloud testbeds properly equipped for Security & Trust experimentations. To reach this goal, ARMOUR will:

- Enhance two outstanding FIRE testbeds ($>$ 2700nodes; $\sim$500users) with the ARMOUR experimentation toolbox for enabling large-scale IoT Security & Trust experiments.
- Deliver properly experimented, suitably validated and duly benchmarked methods and technologies for enabling Security & Trust in the large-scale IoT.
- Define a framework to support the design of Secure & Trusted IoT applications as well as establishing a certification scheme for setting confidence on Security & Trust IoT solutions.

This chapter will present first ARMOUR IoT security testing and its Model Based Testing (MBT) approach. Then, the Section 7.3 of the chapter presents the ARMOUR methodology for benchmarking security and privacy in IoT. Finally, these outcomes will be highlighted trough the prism of the on-going European wide IoT security certification process, before concluding in Acknowledgements.

7.2 ARMOUR IoT Security Testing

This section defines a common language and ontology to express and share project's objectives and activities. This includes primarily a list of vulnerabilities of IoT systems that will be addressed within the ARMOUR project, selected from the analysis of on-going IoT related security initiatives (NIST [36], oneM2M [37], OWASP IoT [34], GSMA [35]). Based on the analysis of the proposed security experiments, ARMOUR defines its methodology based on four segments of an IoT deployment:

- Devices and data
- Connectivity (Wireless)
- Platforms
- Applications and Services

The ARMOUR security framework defines the security in terms of availability, integrity and confidentiality/privacy, offering solutions and guidelines for each of the four segments and the respective identified elements to be secured (Figure 7.1).

The ARMOUR security framework takes as entry the oneM2M vulnerabilities, threats and risk assessment methodology and enriched it with missing vulnerabilities and threats based on the seven experiments to be conducted within the project.

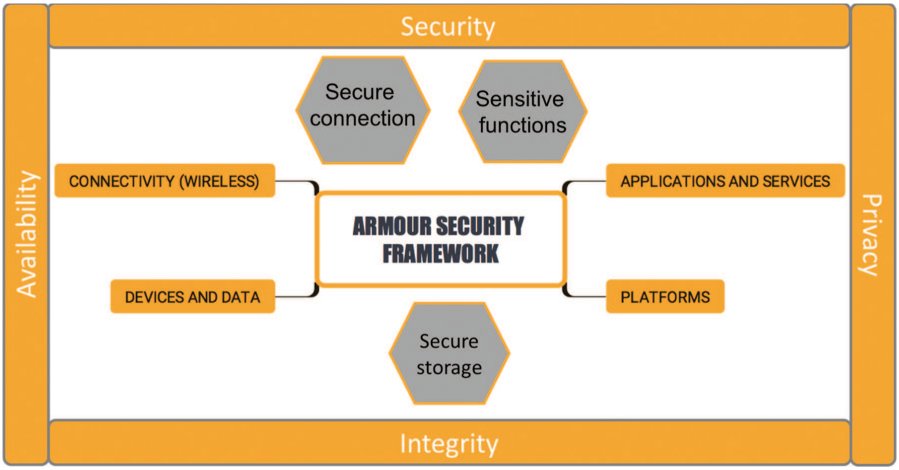


Figure 7.1 ARMOUR Security Framework.

The seven experiments (Figure 7.2) planned within the project and covering the four listed segments, are:

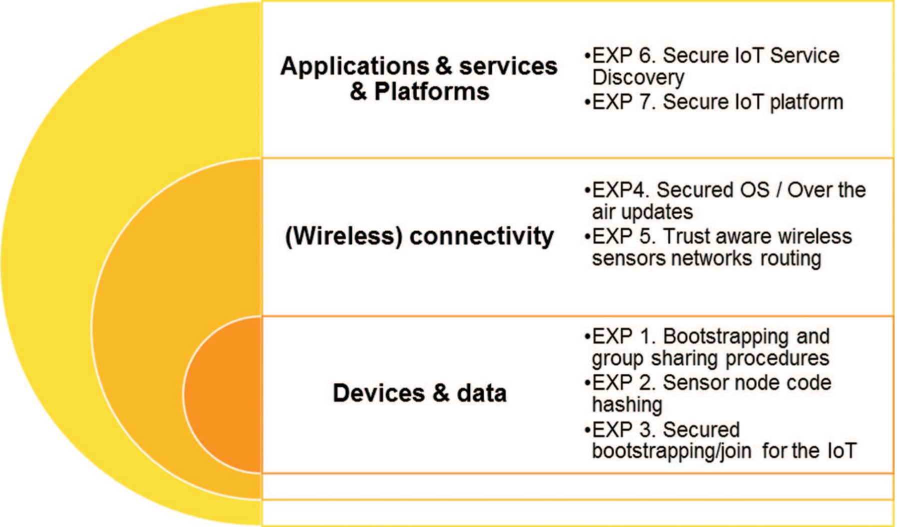

Applications & services & Platforms
•EXP 6. Secure IoT Service Discovery
•EXP 7. Secure IoT platform
(Wireless) connectivity
•EXP4. Secured OS / Over the air updates
•EXP 5. Trust aware wireless sensors networks routing
Devices & data
•EXP 1. Bootstrapping and group sharing procedures
•EXP 2. Sensor node code hashing
•EXP 3. Secured bootstrapping/join for the IoT

Figure 7.2 Positioning of ARMOUR experiments over IoT value chain.

For each experiment, a process is applied to identify the vulnerabilities to be addressed by the experiment and the solutions to be experimented over the ARMOUR testbed to mitigate these vulnerabilities.

This project is driven by the results on the experiments and we deliver in this chapter key elements (vulnerability and test patterns) to provide duly tested and secure solutions in the IoT domain, as well as tools, in particular the ARMOUR Testing and benchmarking methodology and framework towards the creation of an IoT Security labeling and certification system.

7.2.1 ARMOUR Testing Framework

Exisitng Machine to Machine (M2M) standards as well as emerging standards, such as oneM2M, put extreme importance into the definition of security requirements related to security functions in addition to functional requirements.

Moreover, the experiences in security testing and analysis of IoT systems, showed that their Security and Trust will depend on the resistance of the system with respect to:

- Misuses of the security functions,
- Security threats and vulnerabilities,
- Intensive interactions with other users/systems.

Based on the work performed within the project and the testing needs of each experiment, a set of security requirements and vulnerabilities that must be fulfilled by the developed systems have been identified. In order to validate them with respect to the set of requirements, three test strategies have been defined, implementable in combination or individually:

- **Security Functional testing (compliance with agreed standards/ specification):** Aims to verify that system behavior complies with the targeted specification which enables to detect possible security misuses and that the security functions are implemented correctly.
- **Vulnerability testing (pattern driven)**: Aims to identify and discover potential vulnerabilities based on risk and threat analysis. Security test patterns are used as a starting point, which enable to derive accurate test cases focused on the security threats formalized by the targeted test pattern.
- **Security robustness testing (behavioral fuzzing):** Compute invalid message sequences by generating (weighted) random test steps. It enables to tackle the unexpected behavior regarding the security of large and heterogeneous IoT systems.

Model-Based Testing (MBT) approaches have shown their benefits and usefulness for systematic compliance testing of systems that undergo specific standards and that define the functional and security requirements of the system.

ARMOUR proposes a tailored MBT automated approach based on standards and specifications that combines the above mentioned three test strategies built upon the existing CertifyIt technology [16] and TTCN-3 [19] for test execution on the system under test (SUT) into one toolbox called ARMOUR Model Based Security Testing Framework. On the one hand, the CertifyIt technology has already proven its usefulness for standard compliance testing of critical systems, for instance on GlobalPlatform smartcards. Thus, building the ARMOUR approaches upon CertifyIt will allow to get the benefits of a proven technology for conformance testing and introducing and improving it in the domain of IoT. On the other hand, Testing and Test Control Notation version 3 (TTCN-3) is a standardized test scripting language widely known in the telecommunication sector. It is used by the third Generation Partnership Project (3GPP) for interoperability and certification testing, including the prestigious test suite for Long Term

Evolution (LTE)/4G terminals. Also, the European Telecommunication Standards Institute (ETSI), the language's maintainer, is using it in all of its projects and standards' initiatives, like oneM2M. Finally, this testing framework will be deployed within the ARMOUR test beds (FIT IoT lab and FIESTA), for large-scale testing and data analysis.

Based on the evaluation of the testing needs, three possible levels of automation have been identified: The ARMOUR MBT approach with automated test conception and execution based on TPLan tests and TTCN-3 scripts, manual TPLan and TTCN-3 conception and their automated execution on the SUT and finally in-house approaches for testing.

We summarize the approach and illustrate the ARMOUR MBT Security Testing Framework in Figure 7.3. It depicts the three kinds of approaches considered in ARMOUR based on the experiments needs as discussed previously: Tailored MBT approach, manual conception and in-house approach.

The MBT approach in ARMOUR Model Based Security Testing Framework relies on MBT models, which represent the structural and the behavioral part of the system. The structure of the system is modeled by UML class diagrams, while the systems behavior is expressed in Object Constraint Language (OCL) pre- and post-conditions. Functional tests are obtained by applying a structural coverage of the OCL code describing the operations of the SUT (functional requirements). This approach in the

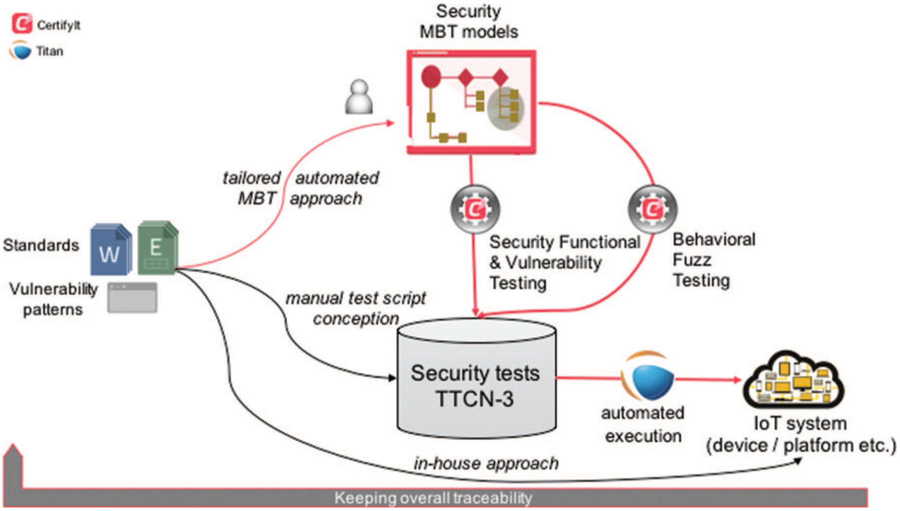


Figure 7.3 ARMOUR Model Based Security Testing Framework.

context of security testing is complemented by dynamic test selection criteria called Test Purposes that make it possible to generate additional tests that would not be produced by a structural test selection criterion, for instance misuse of the system (Model-Based Security Functional Testing) and vulnerability tests, trying to bypass existing security mechanisms (Model-Based Vulnerability Testing). These two approaches generate a set of test cases stored inside a database and then executed on the system. To the difference of them, robustness testing in our context, based on the same model, will generate randomly a test step based on the same MBT model by exercising different and unusual corner cases on the system in a highly intensive way, thus potentially activating an unexpected behavior in the system.

7.2.2 Identified Security Vulnerabilities and Test Patterns

ARMOUR project proposes a security framework based on risk and threat analysis that defines a list of potential vulnerabilities for the different IoT layers (Table 7.1). Each vulnerability is associated to a test pattern. To define the test patterns for each vulnerability, a set of test procedures have been conceived for verifying its system resistance to the vulnerabilities of its experiments. Based on these test procedures specific for the ARMOUR experiments that are introduced in the following section, we have created a library of test patterns generalized to the four IoT segments. The following Figure 7.4. illustrates the methodology applied for the definition

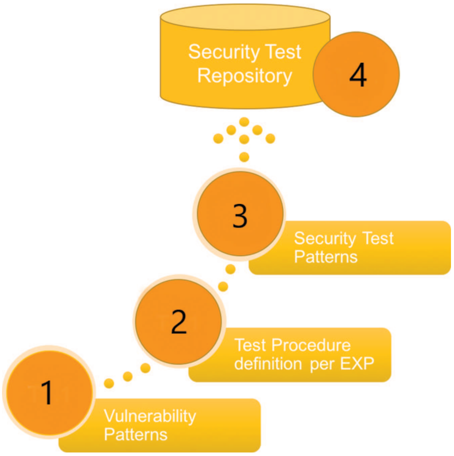


Figure 7.4 Security Test Patterns definition methodology.

Table 7.1 List of defined vulnerabilities

Id	Title
V1	Discovery of Long-Term Service-Layer Keys Stored in M2M Devices or M2M Gateways
V2	Deletion of Long-Term Service-Layer Keys stored in M2M Devices or M2M Gateways
V3	Replacement of Long-Term Service-Layer Keys stored in M2M Devices or M2M Gateways
V4	Discovery of Long-Term Service-Layer Keys stored in M2M Infrastructure
V5	Deletion of Long-Term Service-Layer Keys stored in M2M Infrastructure equipment
V6	Discovery of sensitive Data in M2M Devices or M2M Gateways
V7	General Eavesdropping on M2M Service-Layer Messaging between Entities
V8	Alteration of M2M Service-Layer Messaging between Entities
V9	Replay of M2M Service-Layer Messaging between Entities
V10	Unauthorized or corrupted Applications or Software in M2M Devices/Gateways
V11	M2M System Interdependencies Threats and cascading Impacts
V12	M2M Security Context Awareness
V13	Eaves Dropping/Man in the Middle Attack
V14	Transfer of keys via independent security element
V15	Buffer Overflow
V16	Injection
V17	Session Management and Broken Authentication
V18	Security Misconfiguration
V19	Insecure Cryptographic Storage
V20	Invalid Input Data
V21	Cross Scripting

of generalized security test patterns (3) and towards the conception of security test cases (4).

ARMOUR security test patterns define test procedures for verifying security threats of IoT systems, representatives of different IoT levels, thus facilitating the reuse of known test solutions to typical threats in such systems.

Table 7.2 presents the test patterns and gives an overview of the vulnerabilities that they cover.

With the test vulnerabilities and test patterns defined, each of the testing use cases should generate a set of test cases in order to assess their results. The global overview of the testing methodology is presented in the next section.

7.2.3 ARMOUR IoT Security Testing Approach

The overall ARMOUR test environment for IoT security testing is the following:

Table 7.2 Vulnerabilities overview

Test Pattern ID	Test Pattern Name	Related Vulnerabilities
TP_ID1	Resistance to an unauthorized access, modification or deletion of keys	V1, V2, V3, V4, V5
TP_ID2	Resistance to the discovery of sensitive data	V6
TP_ID3	Resistance to software messaging eavesdropping	V7
TP_ID4	Resistance to alteration of requests	V8
TP_ID5	Resistance to replay of requests	V9
TP_ID6	Run unauthorized software	V10
TP_ID7	Identifying security needs depending on the M2M operational context awareness	V12
TP_ID8	Resistance to eaves dropping and man in the middle	V13
TP_ID9	Resistance to transfer of keys via of the security element	V14
TP_ID10	Resistance to Injection Attacks	V16
TP_ID11	Detection of flaws in the authentication and in the session management	V17
TP_ID12	Detection of architectural security flaws	V18
TP_ID13	Detection of insecure encryption and storage of information	V19
TP_ID14	Resistance to invalid input data	V20

The first step after vulnerability and test pattern identification, is to build a Model Based Testing model representing the behavior of the System Under Test (SUT). The model takes as input a set of security test patterns in order to generate security tests in TTCN-3 format. Next, the tests are compiled and it produces an Abstract Test Suite (ATS) in TTCN-3 format. In order to have an executable ATS, we need to use a compiler that will transform the TTCN-3 code in an intermediate language, like C++ or Java. This is done on purpose because the TTCN-3 code is abstract in a sense that it doesn't define the way to communicate with the SUT or how the TTCN-3 structures will be transformed to a real format. This is a task for the System Adapter and the Codec. There are few commercial and non-commercial compilers available for download from the official TTCN-3 webpage. Usually, all of the compilers follow this procedure:

- Compile TTCN-3 to a target language (Titan uses C++, TTWorkbench uses Java),
- Add the Codec and System Adapter (SA) written in the target language.

In the scope of ARMOUR, Titan test case execution tool was used. The goal of the testing process, as described in the preceding Figure 7.5. ARMOUR overall test environment is to execute the MBT-generated tests on the SUT.

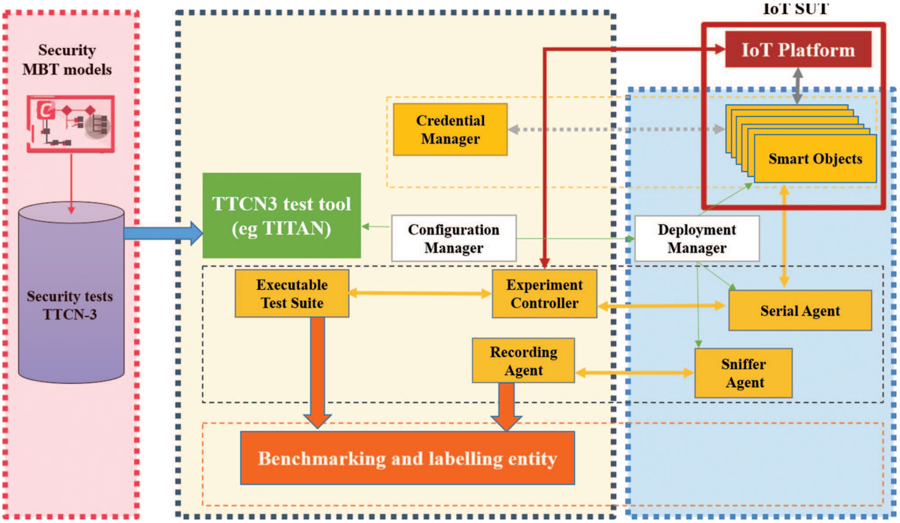


Figure 7.5 ARMOUR overall test environment.

System under test (SUT) refers to a system that is being tested for correct operation. According to ISTQB it is the test object. The term is used mostly in software testing. A special case of a software system is an application which, when tested, is called an application under test. The SUT can be anything involved in an IoT deployment for example:

- Secure communication
- IoT platform
- Device software
- Device Hardware

In ARMOUR, two types of SUT are taken into consideration, IoT Platform and IoT smart objects (Devices). All tests produced are here to stimulate the SUT's and get results in order to assert them and setup a label.

We use for EXP7 an OM2M IoT platform as SUT for the test system. OM2M is an oneM2M standard implementation. The choice of OM2M was made with regards that it is open source and accessible via Eclipse repositories. For a general usage of ARMOUR methodology, any IoT platform implementation is compatible.

SUT can also be called TOE, a Target of Evaluation, a term coming from the Common Criteria. It is defined as a set of software, firmware and/or hardware possibly accompanied by guidance. While there are cases where a

TOE consists of an IT product, this need not be the case. The TOE may be an IT product, a part of an IT product, a set of IT products, a unique technology that may never be made into a product, or a combination of these.

As for the devices tested in ARMOUR, they are devices made available by the FIT IoT- LAB. The test environment is composed of 2 main parts:

I) A laboratory test environment

The first part of the test environment is the laboratory environment. It is composed of many software applications that must be made available online on a cloud machine or other physical address.

- **Credential Manager**: The authority that distributes Cipher keys or certificates.
- **TTCN-3 test tool** (eg TITAN), where the automated tests are stored, compiled and run on the SUT via the Experience controller (Shown in Figure 7.5).
- **Experience Controller** (EC) is a bridge between the Titan tests and FIT IoT lab test environment. It is adaptable to accept other testbeds.
- **Recording Agent** or **sniffer** must be locally installed on the testbed in order to listen to nodes messages. The recorded messages are made accessible online on the cloud.

II) A large-scale test bed: The IoT-LAB

IoT-LAB provides a large-scale test bed. It provides full control of network nodes and direct access to the gateways to which nodes are connected, allowing researchers to monitor nodes energy consumption and network-related metrics, e.g. end-to-end delay, throughput or overhead. The facility offers quick experiments deployment, along with easy evaluation, results collection and analysis. Defining complementary testbeds with different node types, topologies and environments allows for coverage of a wide range of real-life use cases.

IoT-LAB testbeds are located in six different sites across France which gives forward access up to 2700 wireless sensors nodes: Inria Grenoble (928), Inria Lille (640), ICube Strasbourg (400), Inria Saclay (307), Inria Rennes (256) and Institut Mines-Télécom Paris (160) (Numbers as of May 2017).

The IoT-LAB hardware infrastructure consists of a set of IoT-LAB nodes. A global networking backbone provides power and connectivity to all IoT-LAB nodes and guaranties the out of band signal network needed for command purposes and monitoring feedback. This test bed facilitates the availability and implementation of devices in order to run the security tests in a large-scale environment.

7.2.4 Large Scale End to End Testing

Internet of Things (IoT) applications can be found in almost all domains, with use cases spanning across areas such as healthcare, smart homes/ buildings/cities, energy, agriculture, transportation, etc.

It is impossible to provide an exhaustive list of all application domains of the IoT. IoT systems involved in a solution has in most common cases a wide spread of different components such as Hardware (Devices, Sensors, Actuators. . .), Software and Network protocol. ARMOUR project introduces different experiences to test the different components. More specifically, each experience proposes individual security tests covering different vulnerability patterns for one or more involved components. In ARMOUR, EXP1 covers the security algorithms for encryption/decryption and protocols for secure communication. On the other hand, EXP7 covers the IoT data storage and retrieval in an IoT platform. Both experiences can wisely be integrated together. Indeed, it makes sense to pair the secure transfer and storage of data with an IoT platform thus making sure that the confidentiality and integrity of the data is maintained from the moment it is produced by a device, until it is read by a data consumer.

The Model for End-to-End security is conceived as follows:

A wise integration of the experiments is made from different experience models. We call it an Integration model which is used in combination of security vulnerability scenarios in order to generate Abstract Test Cases (Figure 7.6).

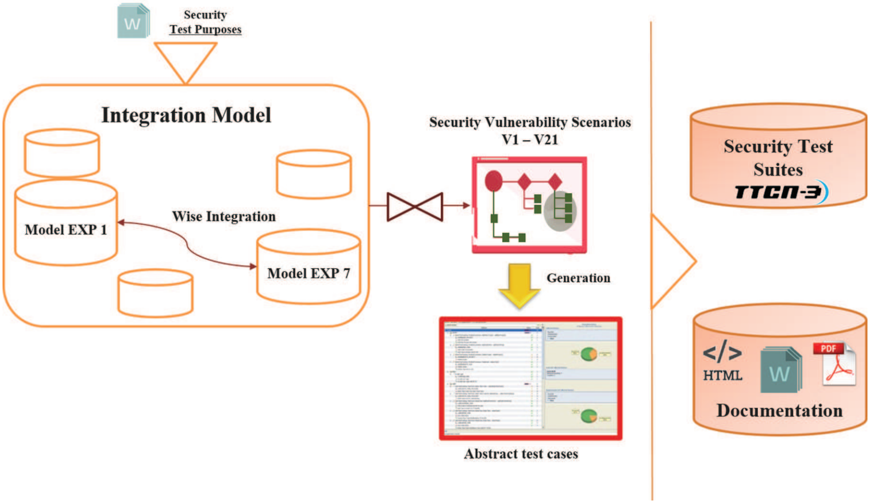


Figure 7.6 End-to-End security description.

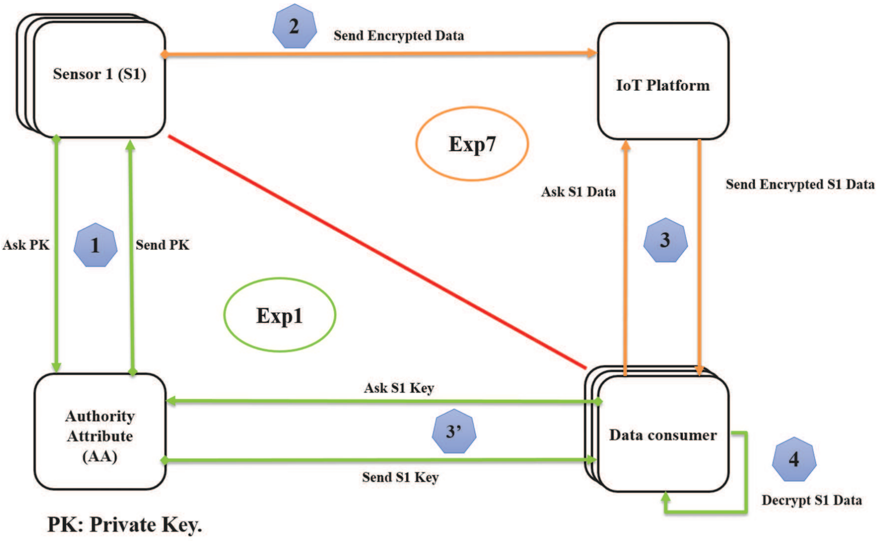


Figure 7.7 Large-Scale End-to-End scenario.

The test cases then follow the normal MBT procedure that is test publication as executables for test automation or as documentation for manual testing.

The End-to-End scenario (E2ES) is described in the above Figure 7.7.

Both data consumer/producers retrieve keys from a credential manager and the IoT platform role is to support the data storage/retrieval. However, a single security problem in any of the components involved in this scenario can compromise the overall security. There is a perception that end-to-end security is sufficient as a security solution, and that network-based security is obsolete in the presence of end-to-end security. In practice, end-to-end security alone is not sufficient and network-based security is also required. This demonstrates that the End-to-End security scenario is here to complete the experience's tests and is not an exhaustive testing method.

7.3 ARMOUR Methodology for Benchmarking Security and Privacy in IoT

Moving from isolated IoT domains to real large-scale deployments requires a better understanding and consensus about actual implications of envisioned infrastructures. For example [1]:

- Is the deployment doing what it is supposed to do?
- Is it the best way to achieve the expected goal?
- Is the deployment creating unexpected issues?
- Is the deployment achieving the expected economic performance?

Providing answers to these questions is of interest to people involved in IoT deployments to identify good practices, avoid traps and make good choices. In this sense, the benchmarking of security and trust aspects is crucial to guarantee the success of large-scale IoT deployments.

Benchmarking IoT deployments should meet different objectives. On the one hand, as we are still in a disharmonized landscape of technologies and protocols, for experimenters and testers, having the whole vision of difficulties and opportunities of IoT deployments is quite difficult. Consequently, there is a need to follow the deployment process, to understand how deployment works, and which security and trust technologies are involved, as well as the impact of a security flaws related to such technologies. On the other hand, and also related with such fragmented landscape of security solutions for the IoT, large-scale deployments will be based on a huge amount of different technologies and protocols. Therefore, the identification of common metrics is crucial to be able to assess and compare such solutions in order to identify good security and trust practices and guidelines.

Towards this end, one of the main goals of the project stems from the need to identify a suitable benchmarking methodology for security and trust in IoT, as a baseline for the certification process. This methodology, as shown in next sections is based on the identification of different metrics associated to different functional blocks, in order to benchmark security and privacy on the different experiments designed in ARMOUR. Specifically, micro-benchmarks provide useful information to understand the performance of subsystems associated with a smart object. Furthermore, they can be used to identify possible performance bottlenecks at architecture level and allow embedded hardware and software engineers to compare and asses the various design trade-offs associated with component level design. Macro-benchmarks provide statistics relating to a specific function at application level and may consist of multiple component level elements working together to perform an application layer task.

The ARMOUR project is focused on carrying out security testing and certification in large-scale IoT deployments. Between both processes, benchmarking is intended to provide different results from testing to serve as the baseline for the certification scheme. Towards this end, for both micro-benchmarking and macro-benchmarking, this methodology will be based on the identification of different metrics per functional block (e.g. authentication). Benchmarking results will help to increase the trust level on the assurance of security properties in IoT products and solutions. Based on this, the next section provides a description of the ARMOUR benchmarking methodology.

7.3.1 Approach Overview

A high-level overview of the proposed benchmarking methodology is shown in Figure 7.8, depicting, the methodology comprising five main stages:

- Experiment design. A concrete experiment is defined in the context of a specific IoT application scenario or use case.
- Test design. The description of the experiment is used to identify different threats and vulnerabilities that can be tested over it. These vulnerabilities and threats are identified from a set of already established vulnerability patterns, and used as an input to generate test patterns in order to define the procedure to test a specific threat.
- Test generation. Based on test patterns, a test model formalizes a specific subset of the functionality of the experiment. Using these models, a set of test suites are generated to be then executed.
- Test execution. From the set of test suites, a set of test adapters are defined to guarantee suites can be executed under different environments and conditions. Specifically, in the context of the project, these tests are intended to be executed in both in-house scenarios and large-scale setting, through the use of FIT IoT-Lab[1].

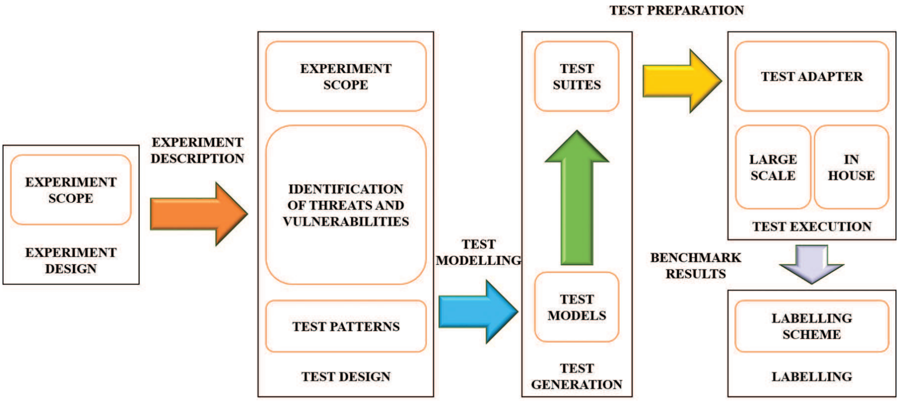


Figure 7.8 ARMOUR benchmarking methodology overview [23].

[1] www.iot-lab.info

- Labelling. The results from the execution of these tests are the used during the certification process by making use of a security labelling scheme.

In following sections, a more detailed overview of these stages is provided.

7.3.2 Experiment Design

The ARMOUR project considers a large-scale experimentally-driven research approach due to IoT complexity (high dimensionality, multi-level interdependencies and interactions, non-linear highly-dynamic behavior, etc.). This research approach makes possible to experiment and validate research technological solutions in large-scale conditions and very close to real-life environments. As a baseline for the benchmarking methodology, different experiments have been designed to test different security technologies, protocols and approaches that are intended to be deployed in typical IoT scenarios (introduced as well in Figure 7.2):

- EXP 1: Bootstrapping and group sharing procedures
- EXP 2: Sensor node code hashing
- EXP 3: Secured bootstrapping/join for the IoT
- EXP 4: Secured OS/Over the air updates
- EXP 5: Trust aware wireless sensors networks routing
- EXP 6: Secure IoT Service Discovery
- EXP 7: Secure IoT platforms

This set of experiments constitutes the basis to obtain benchmarking results from the methodology application to such scenarios. In particular, these experiments have been designed to test specific security properties of technologies and protocols, such as the Datagram Transport Layer Security (DTLS) [2], as the main security binding for the Constrained Application Protocol (CoAP) [3]. These experiments also involve the use of different cryptographic schemes and libraries that will be tested, in order they can be certified according to benchmarking results. By considering different security aspects, during this stage, following a similar approach to the EALs in Common Criteria [4], a risk analysis is performed regarding the assurance level for the environment where the device will operate.

The design and description of such experiments is based on the definition of different required components and testing conditions, to ease the test design, modelling and execution that are defined in the following subsections.

7.3.3 Test Design

During test design, experiments are set up and prepared through the specification of security test patterns that describe the testing procedures. By considering the set of different experiments, as well as the security protocols and technologies involved in each of them, this stage is focused on the definition of different tests to assess the fulfilment of specific security properties on those experiments. These properties are:

- **Authentication**. The devices should be legitimate.
- **Resistance to replay attacks**. Intermediate node can store a data packet and replay it at a later stage. Thus, mechanisms are needed to detect duplicate or replayed messages.
- **Resistance to dictionary attacks**. Intermediate node can store some data packets and decipher them by performing a dictionary attack if the key used to cipher them is a dictionary word.
- **Resistance to DoS attacks**. Several nodes can access to the server at the same time in order to collapse it. For this reason, the server must have DoS protection or a fast recovery after this attack.
- **Integrity**. Received data are not tampered with during transmission; if this does not happen, then any change can be detected.
- **Confidentiality**. Transmitted data can be read only by the communication endpoints.
- **Resistance to MITM (man-in-the-middle) attacks**. The attacker secretly relays and possibly alters the communication. The endpoints should have mechanism to detect and avoid MITM attacks.
- **Authorization**. Services should be accessible to users who have the right to access them.
- **Availability**. The communication endpoints should always be reached and should not be made inaccessible.
- **Fault tolerance**. Overall service can be delivered even when a number of atomic services are faulty.
- **Anonymization**. Transmitted data related to the identity of the endpoints should not be sent in clear.

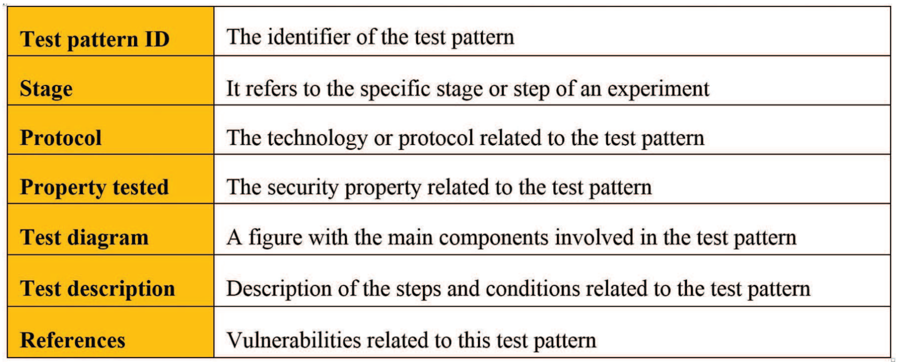
Such properties have been extracted from some of the most referenced security aspects that can be found in current IoT literature [5–9]. These properties can be security properties, such as authentication and integrity, or resistances to certain attacks, such as the MITM and the replay attack.

The proposed methodology is based on the use of these properties and the set of identified vulnerabilities in ARMOUR D1.1 [10] (Table 7.1) that are extracted from oneM2M vulnerabilities [11]. From these properties and the set of vulnerabilities that are described, different tests can be designed in order to prove previous experiments are actually satisfying the set (or a subset) of security properties.

It should be pointed out that the identified security properties are intended to serve as a starting point to build a more generic, stable and holistic approach for security certification in IoT scenarios. This set will evolve during the project to cover other security vulnerabilities.

From these security vulnerabilities and properties, security test patterns are identified to define the testing procedure for each of them. In the context of the project, D2.1 [12] (Table 7.2) already identifies a set of test patterns to be considered for the seven experiments. With all this information, tests patterns can be designed by associating security vulnerabilities and properties. These tests are defined following the schema of the Figure 7.9, which includes a description for each field. The main part of the definition of the test is the test description that must include the steps that we have to follow to make the test and when the test is satisfactory or not.

The proposed template is intended to identify the association or relationship between the test patterns with the security properties and vulnerabilities. The identification of such relationship aims to foster the understanding of the benchmarking methodology approach towards a security certification scheme for the IoT.



Test pattern ID	The identifier of the test pattern
Stage	It refers to the specific stage or step of an experiment
Protocol	The technology or protocol related to the test pattern
Property tested	The security property related to the test pattern
Test diagram	A figure with the main components involved in the test pattern
Test description	Description of the steps and conditions related to the test pattern
References	Vulnerabilities related to this test pattern

Figure 7.9 Test pattern template associating security vulnerabilities and properties [23].

7.3.4 Test Generation

From the test patterns that are designed in the previous stage, during test generation, real tests are defined in order to validate different security properties. For testing purposes, different strategies can be employed; indeed, software testing is typically the process for verifying different requirements are fulfilled regarding the expected behavior of a System Under Test (SUT). In this sense, software security testing [13] is intended to validate that security requirements are satisfied on a specific SUT. According to [14], security testing techniques can be deployed according to their test basis within the secure software development lifecycle into four different types:

- Model-based security testing that is related to the requirements and design models that are created during the analysis and design.
- Code-based testing, which is focused on source and byte code created during development.
- Penetration testing and dynamic analysis on running systems, either in a test or production environment.
- Security regression testing performed during maintenance.

For the proposed methodology, ARMOUR is based on the use of the Model-Based Testing (MBT) approach [15] (as the generalization of Model-Based Security Testing (MBST)). MBT is mainly based on the automatic generation of test cases from a SUT, so consequently, MBST aims to validate security properties with a model of the SUT, in order to identify if such properties are fulfilled in the model.

7.3.5 Test Execution

Once suitable tests are generated, they must be adapted [20] in order to be executed on different environments with its own interfaces and behavior. In the context of the project, generated tests can be executed on an in-house or external large-scale testbed. Both testbed approaches aim to serve as platforms to generate benchmark results from the execution of security tests. In particular, during test execution, following tasks can be carried out [21]:

- Measure, in order to collect data from experiments.
- Pre-process, for obtaining "clean" data to ease assessment and comparisons among different security technologies and approaches.
- Analyse to get conclusions from benchmarking results as a previous step for the labelling and certification.
- Report, informing the inferred conclusions from experiments results.

As an in-house environment example, PEANA is an experimentation platform based on OMF/SFA. An experimenter intended to use this platform will, in first place, use the web portal to schedule and book its experiment. After booking the required components for his experiment, the platform will allow him to access the Experiment Controller, via SSH. This is the place where the experiment will take place. This way, the experimenter must define his experiment using the OMF Experiment Description Language (OEDL) [22] syntax indicating which are the Resource Controllers to be used, i.e. the entity responsible for managing the different scheduled IoT devices. Such definition can also include the new firmware to be tested, and the gathering of returned information. After executing the experiment, the experimenter will be allowed to analyse the performance of his experiment according to the obtained results.

7.3.6 Labelling

During this last stage of the methodology, benchmarking results from the previous stage are used as an input for labelling and certification purposes. The main purpose of this stage is to check if theoretical results that were expected during initial stages are obtained after test executions.

It implies the design a labelling scheme specifically tailored to IoT security and trust aspects, so different security technologies and approaches can be compared, to certify different security aspects of IoT devices. The establishment of this scheme is key to increase trust of IoT stakeholders for large-scale deployments. However, labelling approaches for IoT need to consider the dynamism of IoT devices during their lifecycle (e.g. operating on changing environments), which makes security certification a challenging process.

As an initial approach of labelling for IoT security, below we provide a description of the main aspects that are currently being considered. These aspects will be enhanced within next deliverables to build a solid labelling approach for security aspects in IoT environments. For this approach, it should be noted that labelling must be taken into account the context of the scenario that is being tested and the certification execution. For this reason, and based on Common Criteria (CC) approach, three mains aspects will be considered to be included in the label:

- TOE (Target of Evaluation): In CC, a TOE is defined as a set of software, firmware and/or hardware possibly accompanied by guidance. In this case, a TOE is defined as a context, for example health, industry or home automation. The TOE includes all the devices of the context.

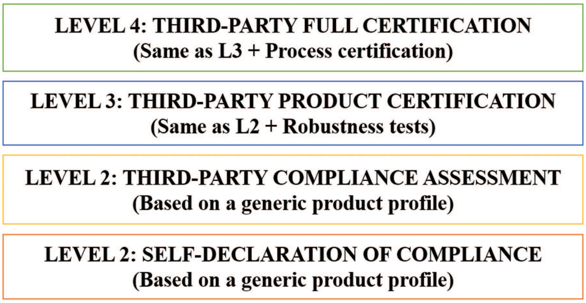
For example, in home automation, the TOE could be the fire sensor, the bulbs, the washing machine, etc.

- Profiles (level of protection): Low, medium and high. The level of protection is related to the threats that can be avoided in the tested scenario.
- Certification execution: The proposed certification execution has 4 levels, which are shown in Figure 7.10. This aspect is intended to be further extended by the certification approach.

In order to be able to label a scenario, we have to consider some metrics that are going to be associated to the execution of the tests, such as detectability, impact, likelihood (that includes difficulty, motivation and benefits), difficulty in obtaining a valid message, percentage of the communication protected with integrity, recoverability, percentage of requests per second necessary, sensibility of the data, time, key length, the facility to hack the server, etc.

In this way, based on different metrics, we have different levels of security associated to a mark for a subset of the security properties that are identified in Section 7.3.3. This association between security properties and marks is shown in Figure 7.11.

As already mentioned, the context must be taken into account in the labelling process. For this reason, we consider an additional parameter: Risk. This parameter is the product of Likelihood and Impact. Likelihood is related to the probability of happening, taking into account the benefit that a person could obtain hacking it, whereas impact is related to the damage that is produced in the scenario in terms of money, sensitive data filtered, scope, etc. These parameters can be variable for each vulnerability. The levels of each one is shown in the Figure 7.12.



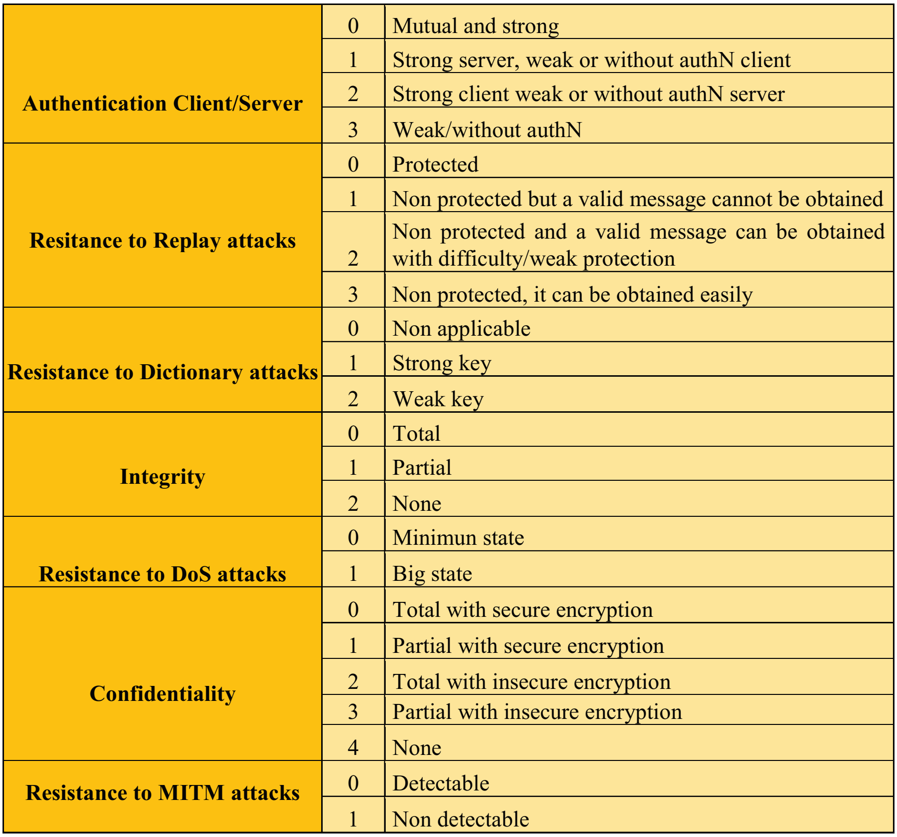
LEVEL 4: THIRD-PARTY FULL CERTIFICATION
(Same as L3 + Process certification)

LEVEL 3: THIRD-PARTY PRODUCT CERTIFICATION
(Same as L2 + Robustness tests)

LEVEL 2: THIRD-PARTY COMPLIANCE ASSESSMENT
(Based on a generic product profile)

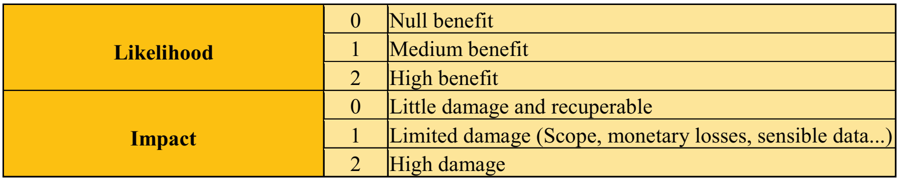
LEVEL 2: SELF-DECLARATION OF COMPLIANCE
(Based on a generic product profile)

Figure 7.10 Certification execution levels [23].



Authentication Client/Server	0	Mutual and strong
	1	Strong server, weak or without authN client
	2	Strong client weak or without authN server
	3	Weak/without authN
Resitance to Replay attacks	0	Protected
	1	Non protected but a valid message cannot be obtained
	2	Non protected and a valid message can be obtained with difficulty/weak protection
	3	Non protected, it can be obtained easily
Resistance to Dictionary attacks	0	Non applicable
	1	Strong key
	2	Weak key
Integrity	0	Total
	1	Partial
	2	None
Resistance to DoS attacks	0	Minimun state
	1	Big state
Confidentiality	0	Total with secure encryption
	1	Partial with secure encryption
	2	Total with insecure encryption
	3	Partial with insecure encryption
	4	None
Resistance to MITM attacks	0	Detectable
	1	Non detectable

Figure 7.11 Association between security properties and marks based on metrics [23].



Likelihood	0	Null benefit
	1	Medium benefit
	2	High benefit
Impact	0	Little damage and recuperable
	1	Limited damage (Scope, monetary losses, sensible data...)
	2	High damage

Figure 7.12 Marks for likelihood and impact parameters [23].

Once we have given each threat and scenario a score (risk and mark), we can define the TOE and the profiles, specifying the minimum level of security they must have, the minimum score (in terms of risk and mark) for each property. If the scenario we are testing achieves the level of security of several profiles, it will be labelled with the major level of security. This implies that an upper level satisfies the requirements of the lower levels.

7.4 A European Wide IoT Security Certification Process

7.4.1 Needs for a European Wide IoT Security Certification Process

On the basis of the concepts described in the previous Sections 7.2, 7.3 and this section, we describe a potential framework for IoT security certification at European level, which is able to support the testing of security and privacy requirements and address the limitations of Common Criteria.

The need for a harmonized security European certification scheme has already been suggested by various studies including [24] and [25], where it is pointed out that current national certification schemes (e.g., Germany, UK and France) could form the basis to create a European certification scheme based on a common approach.

A common European certification scheme would bring a higher level of cyber-confidence to industry buyers and users. A harmonized IoT certified device and product at European level could become an added value in the market. As described in [24], a harmonized security EU certification may be more widely recognized as an international 'quality label' and, hence, support the international competitiveness of European producers. Meanwhile, non-European producers that obtained the same European certification would benefit in an equal way from this 'quality label'.

At the same time, the limitations of existing security certification process like Common Criteria should be addressed. These limitations have been already identified in literature and they are briefly summarized here:

- *Re-certification and patching*. Re-certification of an already certified system or product is an issue raised in [26] and [27] for Common Criteria. Security certification is usually done against a static version of the product and its operative environment. As described before, the IoT environment is especially characterized by dynamic changes of the product and its configuration (in some cases due to the need to install security patches). We note that some European countries like France have already proposed alternative approaches for security certification like the Certification de Sécurité de Premier Niveau (CSPN) described in [28].
- *Lack of mutual recognition*. Security certification profiles and testing defined in some European countries may not be equivalent to security profiles in other countries. The lack of mutual recognition is disruptive to the European single digital market, because consumers of security certified products will not be able to compare the levels of security

certification. Existing European organizations like SOG-IS (Senior Officials Group for Information System Security) are already addressing this issue by working on the harmonization of protection profiles at European level.

- *Certification costs*. Common criteria certification is considered a long and expensive process, which does not make it suitable for fast market deployment or relative short product cycles as in the consumer market [29] and [30]. This issue is particularly relevant for IoT products and devices because of the low profit margins for IoT manufacturers.

The proposed European wide framework aims to address and mitigate the previous issues on the basis of the concepts and tools (MBT, TTCN) described in the previous sections of this chapter.

7.4.2 Main Elements of the European wide IoT Security Certification Process

The overall representation of the framework is provided in Figure 7.13. The green elements represent artefacts (e.g., test suites) while the azure rectangles represent the roles. The main roles and elements in Figure 7.13 are described here:

- European Governing Board. This is the main governing body of the overall EU security certification framework. The European Governing board is composed at least by the representatives of the national certification bodies and the European Commission. SOG-IS will also be part of the European Governing Board. The EGB is responsible for drafting and managing changes to the security certification process. The EGB is also responsible for defining and maintaining the benchmarks.
- IoT Product Manufacturer. This is the manufacturer of the product to be submitted for certification. Manufacturers can be present in different domains or a single domain (e.g., road transportation or energy). The IoT product manufacturer is also responsible to express needs, identify vulnerabilities or define security requirements.
- European accreditation bodies and auditors. They are responsible for the accreditation of the certification centres and the periodic auditing.
- The Labelling Program Authority is the European (or member state entity), which is responsible for assigning the labels after a successful certification process. The Labelling Program authority associates the certification environment, test suites, tools, domain and processes to

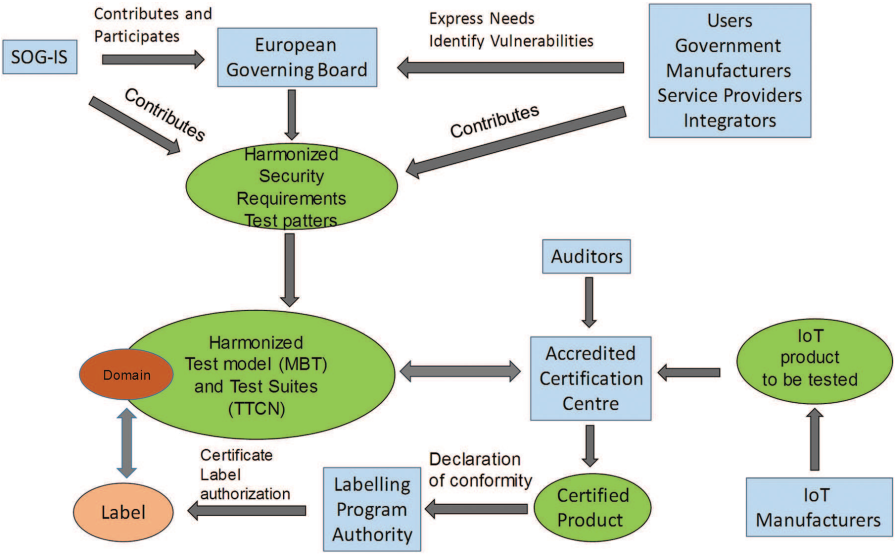

SOG-IS
Contributes and Participates
European Governing Board
Express Needs Identify Vulnerabilities
Users Government Manufacturers Service Providers Integrators
Contributes
Contributes
Harmonized Security Requirements Test patters
Auditors
Harmonized Test model (MBT) and Test Suites (TTCN)
Domain
Accredited Certification Centre
IoT product to be tested
Certificate Label authorization
Declaration of conformity
Label
Labelling Program Authority
Certified Product
IoT Manufacturers

Figure 7.13 Proposed IoT EU Security Certification Framework.

specific labels, which are then provided to the IoT manufacturer after a product submission.

- Accredited certification centres. An accredited certification centre performs the test execution on the basis of the pre-defined harmonized protection profile. Note that existing test beds could be used. A number of firms in Europe have been certified as Commercial Licensed Evaluation Facilities (CLEFs) under the Common Criteria, and do certification work that is recognised across participating states. They mostly evaluate software products for government use, though there is some work on products such as smartcards that are sold into the banking industry.
- Users. They are the users of the certified product. They use the label information as a metric to drive their procurement process. Users can be citizen, public (e.g., government) or private companies. Together with manufacturers, governments, service providers and integrators, they express needs, identify vulnerabilities or define security requirements.
- SOG-IS is the European security organization described before. We consider SOG-IS a very important element of the European security certification framework because of its experience and capabilities. Still, the role and duties of SOG-IS in such a framework must be carefully evaluated together with SOG-IS representatives.

The main flows of operation in the framework are the following:

- The IoT manufacturer submits the application for the security certification of an IoT product.
- Various users (manufacturers, governments, service providers, integrators) express the security requirements and needs to define new test patterns and test suites.
- The European governing board review the applications for new test patterns and security requirements and ensure that they are harmonized at European level.
- From the test patterns and security requirements, test suites (i.e., TTCN) and test models (i.e., MBT) are created for specific categories of IoT products. The entities responsible for creating the test suites and test models is not defined at this stage, but they could be represented by security companies with skills in security certification and drafting of protection profiles from Common Criteria.
- A label is issued by the labelling program authority once that an accredited certification centre has successful certified an IoT product.

7.4.3 Security and Privacy Requirements

One task, which can be performed by the proposed EU IoT security certification framework is to address privacy requirements as well.

The concept of privacy certification is not new, even if security certification (or safety certification) has been historically the main priority. European Commission's General Data Protection Regulation [31] in Recital 77 encourages the "establishment of certification mechanisms, data protection seals and marks" to enhance transparency, legal compliance and to permit data subjects [individuals] the means to make quick assessments of the level of data protection of relevant products and services.

A relevant case study for Privacy certification is the concept of Privacy Seal [32]. The Privacy seal is a trans-European privacy trust mark issued by an independent third party certifying compliance with the European regulations on privacy and data protection. See (see https://www.european-privacy-seal.eu/ by EuroPriSe for more information on the Privacy Seal and the activities carried out by EuroPriSe. The Privacy seal concept is relatively similar to the label concept of security certification where the label is the seal itself.

The overall process to obtain a Privacy Seal could also be similar to envisaged EC security certification process described in the previous section.

Private and public manufacturers of IoT products can apply for the certificate and related European label. The trust mark is awarded after successful evaluation of the product or service by independent experts and a validation of the evaluation by an impartial certification authority.

Reference [32] provides and extensive description of the most common Privacy Certification processes available in the world. One of the main examples is TRUSTe, which defines processes for Privacy certifications for various products and services. In [33] are defined Privacy certification standards for Smart Grids, Enterprise and others. TRUSTe works closely with stakeholders to identify the needs for the definition of new Privacy certification standards. The standards define the Privacy Program requirements, the vendor must satisfy in its service or product. Examples of requirements defined in the TRUSTe standards are related to protection against phishing or the implementation of encryption methods for data protection and data confidentiality.

These examples already show that security certification and privacy certification cannot be disjointed but they should be combined as they often address the same or similar requirements (e.g., access control, confidentiality) or solutions (e.g., cryptographic algorithms).

Figure 7.14 provides a preliminary description on the potential process to support both security and privacy requirements. While applications experts (the various users from the previous subsection) could define the security requirements, the EDPS could work with the EGB and other stakeholders to define the privacy requirements. Both categories of requirements can be used to define the draft test patterns and models. Then the flow could be similar to what already presented in Figure 7.13.

7.4.4 How the EU Security Certification Framework Addresses the Needs

In this section, it is described how the proposed EU security certification framework could address the needs and limitations of common criteria identified in Section 7.4.1.

The *Lack of mutual recognition* is addressed by defining harmonized test patterns, models and suites at European level. While, these test artefacts could be still specific for domain (e.g., specific context like automotive, energy), they would be harmonized at European level for classes of products (e.g., the ITS DSRC system in a vehicle or an IoT Gateway in a smart home).

The *Re-certification and patching* issue is addressed by the use of models and well-defined test suites. By using a common criteria terminology where

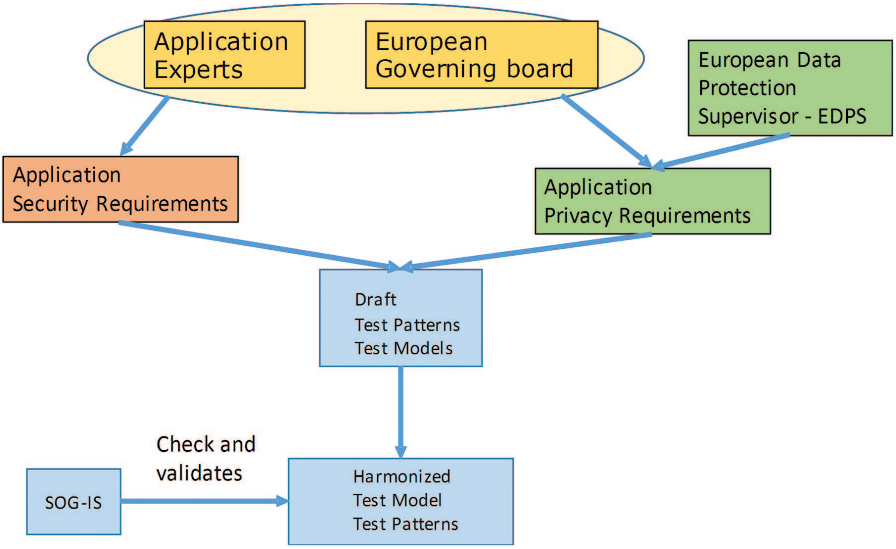


Figure 7.14 Combination of security and privacy requirements.

the TOE is the IoT device to be certified, the ARMOUR project tackles finely the issue on the re-certification and helps the review of security impact using a safe regression testing techniques. Safe regression testing techniques make possible to select test cases for execution relevant for testing the modified TOE without omitting test cases that may reveal faults. The re-evaluation of a TOE is based on a comparison between the initial and the updated version of the TOE's evidence. Several elements being part of the TOE's evidence may evolve:

- The model (the TOE's functional specification has evolved)
- The security patterns (new security properties are tested for the same TOE, for instance to reach a higher label)
- Both may evolve (as the changes in the security test patterns are often due to the development of new functionalities in the TOE)

The test tool based on an automated comparison of the TOE's formal evidence expressed in the form of an MBT model) can detect if there are any impacted security properties. Based on the analysis it will:

- Select a set of existing test cases that are impacted by the updated TOE's evidence
- Generate new tests to cover any new or evolved security test pattern

- Report on impacted vulnerability patterns, which ensure traceability to the security properties part of the security certification.

These output elements of the ARMOUR Testing Framework will serve as facilities to easy and lower the cost of the re-certification. The systematic and automated process offered by the framework will on the one hand easy the decision process on the need for re-certification, as it will report on the impacted security test cases and thus impacted security properties. On the other hand, only a sub-set of the test cases will be executed on the test bed and analysed, which lowers the processing and post processing testing activity.

The *Certification costs* issue is mitigated (not resolved as described below) by the analysis provided above (i.e., use of automated test suites) and by the consideration that a single European combination of test model, pattern and suite is created for categories of IoT devices thus creating a mass market for security certification for specific categories of IoT products.

The trade-off in setting up the EU Security certification framework is the cost of setting up this framework with an increased role for SOG-IS, the creation of the EGB and the definition of new flows and processes. On the other side, these processes and roles (e.g., accredited auditor and security certification lab) do already exist in Europe but they are quite fragmented. The authors of this book chapter believe that the adoption of a common methodology and tools can improve the security and privacy certification process in Europe.

7.5 Conclusion

This book chapter has described the application of Model Based Testing (MBT) for the security certification of IoT products. The application of MBT can improve the efficiency of the IoT security certification process, which is especially important in the IoT context, due to the presence of a wide range of vulnerabilities and the dynamic environment where IoT products are used.

The book chapter does also describe a potential EU-wide security certification framework for IoT security certification, with a clear definition of roles and processes.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688237.

References

[1] PROBE-IT, Deliverable D2.1, 'Benchmarking framework for IoT deployment evaluation', Oct, 2012.

[2] E. Rescorla, N. Modadugu, 'RFC 6347 – Datagram Transport Layer Security Version 1.2.', Internet Engineering Task Force (IETF), Jan, 2012.

[3] Z. Shelby, K. Hartke, C. Bormann, 'RFC 7252 – The Constrained Application Protocol (CoAP)'. Internet Engineering T ask Force (IETF), June, 2014.

[4] The Common criteria, 'Collaborative Protection Profiles (cPP) and Supporting Documents (SD)', 2015.

[5] G. Selander, F. Palombini, K. Hartke, 'Requirements for CoAP End-To-End Security', 2016.

[6] K. Moore, R. Barnes, H. Tschofenig, 'Best Current Practices for Securing Internet of Things (IoT) Devices', 2016.

[7] M. Abomhara, G. Koien, 'Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks', Journal of Cyber Security Vol. 4, pp. 65–88, 2015.

[8] T. Heer, O. Garcia-Morchon, R. Hummen, S. Loong, S. Kumar, K. Wehrle, 'Security Challenges in the IP-based Internet of Things', Wireless Pers Commun, Vol. 61, pp. 527–542, 2011.

[9] H. Suo, J. Wan, C. Zou, and J. Liu, 'Security in the Internet of Things: A Review', International Conference on Computer Science and Electronics Engineering, 2012.

[10] ARMOUR, 'Deliverable. D1.1 – ARMOUR Experiments and Requirements', Aug, 2016.

[11] OneM2M-TR-0008, 'Analysis of Security Solutions for oneM2M System v0.2.1', 2013.

[12] ARMOUR, 'Deliverable D2.1 – Generic test patterns and test models for IoT Security Testing', Aug, 2016.

[13] B. Potter, G. McGraw, 'Software security testing', IEEE Security & Privacy, Vol 2, pp. 81–85, 2004.

[14] M. Felderer, M. Büchler, M. Johns, A. Brucker, R. Breu, A. Pretschner, 'Security Testing: A Survey', Advances in Computers, pp. 101, 1–51, 2016.

[15] M. Utting, B. Legeard, 'Practical model-based testing: a tools approach', 2010.

[16] B. Legeard, A. Bouzy, ‘Smartesting certifyit: Model-based testing for enterprise it’, Software Testing, Verification and Validation (ICST), IEEE Sixth International, pp. 391–397, March, 2013.

[17] J. Botella, F. Bouquet, J. Capuron, F. Lebeau, B. Legeard, F. Schadle, ‘Model-Based Testing of Cryptographic Components’, Software Testing, Verification and Validation (ICST), IEEE Sixth International Conference, pp. 192–201, March, 2013.

[18] J. Warmer, A. Kleppe, ‘The object constraint language: Precise modeling with UML’, Addison-Wesley Object Technology Series, 1998.

[19] C. Willcock, T. Deiß, S. Tobies, F. Engler, S. Schulz, ‘An introduction to TTCN-3’, John Wiley & Sons, 2011.

[20] G. Baldini, A. Skarmeta, E. Fourneret, R. Neisse, B. Legeard, F. Le Gall, ‘Security certification and labelling in Internet of Things’, IEEE 3rd World Forum on Internet of Things (WF-IoT), 2016.

[21] S. Pérez, J.A. Martínez, A. Skarmeta, M. Mateus, B. Almeida, P. Maló, “ARMOUR: Large-Scale Experiments for IoT Security & Trust”, IEEE 3rd World Forum on Internet of Things (WF-IoT), 2016.

[22] T. Rakotoarivelo, M. Ott, G. Jourjon, I. Seskar, ‘OMF: a control and management framework for networking testbeds’, ACM SIGOPS Operating Systems Review, Vol 4, pp. 54–59, 2010.

[23] ARMOUR, ‘Deliverable D4.1 – Definition of the large-scale IoT Security & Trust benchmarking methodology’, 2016.

[24] Security Regulation, Conformity Assessment & Certification Final Report – Volume I: Main Report, Brussels October 2011. http://ec.europa.eu/dgs/home-affairs/e-library/documents/policies/security/pdf/secerca_final_report_volume__1_main_report_en.pdf. Last accessed 4/June/2017.

[25] Proposals from the ERNCIP Thematic Group, “Case Studies for the Cyber security of Industrial Automation and Control Systems”, for a European IACS Components Cyber security Compliance and Certification Scheme, https://erncip-project.jrc.ec.europa.eu/component/jdownloads/send/16-case-studies-for-industrial-automation-and-control-systems/60-proposals-from-the-erncip-thematic-group-case-studies-for-the-cyber-security-of-industrial-automation-and-control-systems-for-a-european-iacs-components-cyber-security-compliance-and-certification-scheme?option=com_jdownloads. Last accessed 4/June/2017.

[26] Minutes of the Joint EC/ENISA SOG-IS and ICT certification workshop. October 2014 ENISA. https://www.enisa.europa.eu/events/sog-is/minutes. Last accessed 4/June/2017.

[27] Kaluvuri, S. P., Bezzi, M., & Roudier, Y. (2014, September). A quantitative analysis of common criteria certification practice. In International Conference on Trust, Privacy and Security in Digital Business (pp. 132–143). Springer International Publishing.

[28] Antoine Coutant. French Scheme CSPN to CC evaluation http://www.yourcreativesolutions.nl/ICCC13/p/CC%20and%20New%20Techniques/Antoine%20COUTANT%20-%20CSPN%20to%20CC%20Evaluation.pdf. Last accessed 4/June/2017.

[29] "Common Criteria Reforms: Better Security Products through Increased Cooperation with Industry", available at: http://www.niap-ccevs.org/cc_docs/CC_Community_Paper_10_Jan_2011.pdf. Last accessed 4/June/2017.

[30] Anderson, R., & Fuloria, S. Certification and evaluation: A security economics perspective. In Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on (pp. 1–7). IEEE. September 2009.

[31] General Data Protection Regulation. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679. Last accessed 4/June/2017.

[32] EU privacy seals project. Inventory and analysis of privacy certification schemes. Rowena Rodrigues, David Barnard-Wills, David Wright. http://bookshop.europa.eu/en/eu-privacy-seals-project-pbLBNA26190/ ISBN: 978-92-79-33275-3. Last accessed 4/June/2017.

[33] TRUSTe Privacy Certification Standards, https://www.truste.com/privacy-certification-standards/ Last accessed 4/June/2017.

[34] OWASP, https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project

[35] GSMA Security Framework CLP11, February 2016.

[36] NIST, CSRC, http://csrc.nist.gov/groups/SMA/fisma/framework.html

[37] "OneM2M security solutions", oneM2M-TR-0008-Security-V1.0.0, 2014.