FutureTPM

D1.1

FutureTPM Use Cases and System Requirements

Project number:	779391
Project acronym:	FutureTPM
Project title:	Future Proofing the Connected World: A Quantum-Resistant Trusted Platform Module
Start date of the project:	1st January, 2018
Duration:	36 months
Programme:	H2020-DS-LEIT-2017

Deliverable type:	Report
Deliverable reference number:	DS-06-779391 / D1.1/ 1.0
Work package contributing to the deliverable:	WP 1
Due date:	Jun 2018– M06
Actual submission date:	2nd July, 2018

Responsible organisation:	S5
Editor:	Minas Pertselakis, Ioanna Michael, Dimitris Panopoulos
Dissemination level:	PU
Revision:	1.0

Abstract:	D1.1 defines the technical requirements of FutureTPM, alongside with the requirements of the use cases. Its purpose to define the parameters for the rest of the FutureTPM project and provide the necessary input to the architecture.
Keywords:	Requirements, Technical Requirements, Use Cases, user Stories, MVP, Vision

The project FutureTPM has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 779391.

Editor

Minas Pertselakis, Ioanna Michael, Dimitris Panopoulos (S5)

Garfield Benjamin, José Moreira, Mark Ryan (UB)

Contributors (ordered according to beneficiary numbers)

Thanassis Giannetsos, Liqun Chen (SURREY)

Daniele Sgandurra, Elizabeth Quaglia (RHUL)

Leonel Sousa, Paulo Martins (INESC-ID)

Roberto Sassu, Silviu Vlasceanu (HWDU)

Fanis Sklinos, Stratos Moros (INDEV)

Alexandros Tsitsanis (S5)

Disclaimer

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The content of this document reflects only the author`s view – the European Commission is not responsible for any use that may be made of the information it contains. The users use the information at their sole risk and liability.

List of Abbreviations

Abbreviation	Translation
BIOS	Basic Input/Output System
CA	Certification Authority
HSM	Hardware SecurityModule
MVP	Minimum Viable Product
NVRAM	Non-volatile Random-Access Memory
PRNG	Pseudo Random Number Generator
TEE	Trusted Execution Environment
TPM	Trusted Platform Module
QR	Quantum Resistant

Executive Summary

D1.1 defines the high level technical requirements of the FutureTPM platform, alongside the requirements of the three use cases that will be used as validation testbeds – namely *Secure Mobile Wallet and Payments, Personal Activity* and *Health Kit Data Tracking and Device Management*. In this context, it provides a detailed view of FutureTPM's reference scenarios and describes a number of user stories (within these scenarios) which are used to better explain the functionality of each use case that will be tested during the project. Out of the three different use cases a total number of 29 user stories has been extracted. Moreover, an initial set of KPIs per use case has been devised (to be revised under WP6, T6.1), hinting at business and technical indicators that will be tested under the piloting work package of the project.

In parallel, the consortium has worked towards defining the technical requirements for the FutureTPM platform, including an analysis of how present-day TPM functionality can be implemented using QR cryptography. Requirements, with a view on a holistic solution that could be offered as a QR TPM have been categorised as *mandatory* and *desirable*, and in total, 52 technical requirements have been extracted (27 mandatory and 25 desirable).

Following these activities, a mapping between the project's technical requirements and the use case requirements has been conducted, identifying the exact needs of each use case and the type of TPM environment that will be tested in each scenario. This resulted to INDEV (leading the Secure Mobile Wallet and Payments use case) building a use case based on hardware TPM, S5 (leading the Personal Activity and Health Kit Data Tracking use case) on software TPM and HUAWEI (leading the Device Management use case) testing the virtual TPM implementation.

That mapping has also allowed the consortium to devise the FutureTPM MVP, which includes a total number of 37 technical requirements which is suggested as the initial requirement package to be considered for implementation by the project. This will cover the needs of the use cases needs as well as other horizontal needs that are necessary for demonstrating the overall FutureTPM concept.

The overall purpose of D1.1 is to provide a reference document for the FutureTPM project to be used as input to the platform's architecture definition, including the needs of the quantum resistant cryptographic primitives, the functionality of the demonstrators and the metrics against which those demonstrators will be assessed.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Towards FutureTPM

The advent of the Internet has been one of the most important technological changes experienced by the society in the last decades. It has a profound impact in the way we communicate, conduct business and socialize. As expected, such a major transformation brought quite a lot of difficult challenges to tackle, for example, in the fields of regulation, standardization, privacy, ethics or economics. Security (and cybersecurity), as an essential factor for the development of any digital technology, soon became one of the fields where the scientific community devoted much of their research efforts [1].It is widely believed that the next major technological transformation that we as a society will experience will come in the next few years with the introduction of the Internet of Things (IoT). The interconnection through the Internet of everyday devices, home appliances and other items embedded with inexpensive electronics (sensors, actuators, connectivity endpoints) will undoubtedly increase the amount of privacy and security challenges, in addition to the already existing ones in the Internet.

A Trusted Platform Module (TPM) is a highly secure hardware component that, together with the BIOS, can serve as a root of trust, that is, a hardware anchor on which secure systems could be built. Its main goals are to provide a protected space for key operations and other security critical tasks, cryptographic functions, measure and report the behaviour of computing platforms, store data securely and perform secure authorization.

It is designed to enhance platform security beyond the capabilities of software and shield cryptographic and sensitive material from software-based attacks. Moreover, augmenting computers with these hardware modules adds powerful functionality in distributed settings, allowing us to reason about the security of these systems in new ways.

The TPM was conceived by a consortium called the Trusted Computing Group (TCG). The first widely adopted version TPM 1.1b was released in 2003, which was subsequently revised to version 1.2, published in about 2005, and was later standardized by the ISO/IEC in 2009 [2].The last major revision of TPM is version 2, released on 2014 and standardized in 2015. The last updates to the TPM 2.0 specification have been made early in 2018. TPM 2.0 has been designed with a "library" approach. This allows vendors to choose TPM functionality for different implementation levels and platforms. Also, new features and functions were added, such as the ability to implement new cryptographic algorithms as needed. This flexibility allows the latest TPMs to be used for many embedded applications, including but not limited to, automotive, industrial, cloud computing and IoT.

1.2 High-level Vision

The overarching vision of the FutureTPM project is to futureproof the TPM as a widely adopted and deployed specification for hardware anchored security. In a constantly shifting technological landscape, incorporating the full range of devices from digital infrastructure to user devices, and including IoT, Industrial Control Systems (ICS), network hardware and software-based trust on smart devices is of paramount importance. Any standardised platform that seeks to fulfil the security needs of these diverse systems for the next twenty years or more will require specific considerations. This includes, perhaps most notably, quantum resistance but also other techniques for enhancing longevity such as agility in which cryptographic protocols are used, a thorough security analysis, demonstrated real-world applications, and the trust of the research community and industry. The FutureTPM project aims to accomplish this vision by *building on the existing strengths and reputation of the TPM while building in assurances to contribute towards the future of security*.

1.3 Scope and Purpose

The goal of the FutureTPM project is to design a Quantum-Resistant (QR) Trusted Platform Module (TPM) by selecting (and designing) and developing QR algorithms suitable for inclusion in a TPM [3]. The algorithm design will be accompanied with implementation and performance evaluation, as well as formal security analysis in the full range of TPM environments: i.e. hardware, software and virtualization environments. The deliverable at hand provides an introduction to the project's scope and vision, denoting the technical requirements that have been identified by the partners, and the use cases for the implementation of the envisaged QR TPM platform. As described in the project's DoA, use cases in online banking, activity tracking and device management will provide environments and applications to validate the FutureTPM framework in various formats, such as hardware, software and virtual TPMs. In order to achieve these goals, it is essential to first understand and systematically document the requirements of the TPM design and its application to the use cases and then map them to the requirements coming from research related to the needs for a QR implementation of TPM.

This deliverable (D1.1) defines the high level technical requirements of the FutureTPM platform, alongside the requirements of the three use cases that will be used as validation testbeds. In this context, it provides a detailed view of FutureTPM's reference scenarios and describes a number of use stories to be investigated within the reference scenarios. It then derives the technical requirements for the FutureTPM project, including an analysis of how present-day TPM functionality can be implemented using QR cryptography. This document defines the parameters for the rest of the FutureTPM project and provide the necessary input to the architecture definition, including the needs of the quantum resistant cryptographic primitives, the functionality of the demonstrators and the metrics against which those demonstrators will be assessed.

1.4 Relation to other WPs and Deliverables

As a technical requirements deliverable for the project as a whole, D1.1 arguably relates (and serves as the basis) to all later WPs and deliverables. Within WP1, this deliverable will be linked directly to *D1.2 FutureTPM Reference Architecture*, in which the consortium will outline the specifications of the overall FutureTPM platform to be implemented towards meeting the requirements defined in this document. In combination with *D1.3 Security Risks in QR Deployments*, the present deliverable will feed into the deliverables of WP2 (*D2.1-2.3 First, Second and Final Reports on New QR Cryptographic primitives*), which will define the underlying technical solutions to be used in converting the quantum-insecure TPM 2.0 into the quantum resistant FutureTPM. The use case descriptions, and accompanying qualitative metrics, will also provide the narrative basis for the specific quantitative metrics to be defined in the risk-assessment frameworks of WP4 and applied throughout WP5 and WP6 in the creation and evaluation of the demonstrators.

1.5 Deliverable Structure

In Chapter 2 we describe the background regarding TPM technologies, discuss their weaknesses in relation to quantum resistance and through this analysis we suggest the novelties that can be brought forward by FutureTPM, presenting the consortium's shared vision of the project. The overall methodology of the deliverable is presented in Chapter 3, where the methodology to derive to the project's MVP and to the technical and business requirements is provided. Chapter 4 outlines the use cases in detail, including descriptions of the systems, the type of TPM adopted by each demonstrator, and the user stories from which the requirements will be drawn. Chapter 5 serves with detailed lists of the technical requirements for both functionality and security. Finally, Chapter 5 maps the user stories to the technical requirements and defines scope/requirements of the project's demonstrator (MVP), and Chapter 6 concludes the deliverable.

Chapter 2 Background

2.1 Analysis of present-day TPM functionalities

With the idea of becoming a hardware security anchor in mind, the TPM was developed to address a number of issues in the fields of security and privacy. Towards this direction, the following functionalities have been considered by the TCG in the current specification (TPM 2.0).

Cryptographic functionalities. The Cryptography subsystem is responsible for implementing the cryptography stack of the TPM. Its main components are described below which provide the basic set of cryptographic primitives supported:

- *Random number generation*. The hardware nature of the random number generator [4] offers a better source of entropy to create cryptographic material, compared to software-based PRNGs. It nominally consists of an entropy collector, a state register, and a mixing function (typically, a hash function). The sources of entropy can be as diverse as noise, clock variations, air movement, and other types of events.
- *Hash functions.* The current specification of the TPM allows the computation of hash functions as a single call for small inputs or as the usual start/update/complete sequence.
- *Message authentication codes.* The TPM implements the HMAC algorithm described in the ISO/IEC 9797-2, again, using the 2 modes of operations described for the hash functions.
- *Asymmetric cryptography.* The TPM uses asymmetric algorithms for attestation, identification and secret sharing. Currently, the only supported asymmetric algorithms are based on RSA and ECC using prime curves. The functionalities provided are signature generation and verification, encryption and decryption. Several padding schemes are permitted for the input data, e.g., PKCS#1, OAEP, and no padding.
- *Symmetric cryptography.* TPMs use symmetric encryption to encrypt data during a number of operations such as authentication or transport sessions, and also to protect data that is stored outside the TPM. The block cipher modes referenced in the current specification are defined in ISO/IEC 10116:2006.
- *Key generation and key derivation.* TPM offers two types of key generation: either from the provided random number generator or derived using a key derivation function and a seed value, depending on the application. For the purpose of key derivation (from another seed value) the KDF used is specified in SP800-108, with HMAC as the pseudo-random function, and in SP800-56A.

The current TPM specification does not enforce implementation of any specific algorithm, and developers are responsible of considering factors such as use cases, best practices, strength and backwards compatibility. However, the TCG specifies sets of algorithms to be incorporated in certain platforms. Moreover, it is required that the strength of, at least, one algorithm set is of 112 bits or more. Also, the specification enforces security measures such as rejection of known weak keys for encryption algorithms, or brute-force attacks.

The specification also incorporates the concept of algorithm agility [5], that is, the ability of algorithms being added or subtracted from the specification without requiring that the entire specification be rewritten. This way, if one cryptographic algorithm is weakened by cryptanalysis in the future, it can be removed and replaced without changing the specification.

Storage capabilities

- *Secure storage.* A TPM has a limited memory to securely store cryptographic objects. To extend this storage, it has also access to a self-generated secret key, so that it can encrypt additional material and store it externally, having a virtually unlimited storage space. Several features such as authentication and integrity checks are added to the protected object that are stored either internally or externally.

- *Non-volatile storage.* The specification also defines a small amount of NVRAM storage, with restricted access-control properties. This is used as a persistent space to store, for example, root keys of certificate chains or to have access to decryption keys used before the hard disk is available. It can be used both to store data structures defined by the TPM specification or unstructured data defined by the user.
- *Platform configuration registers* (PCRs). PCRs [6], [7], are protected memory registers for storing integrity measurements or platform software state. This allows secure storage and reporting of security-relevant data (unauthorized changes to the BIOS, possible root-based modifications, boot-sector changes, etc.). Even though the memory used by the PCRs is limited, they can be used to represent an unlimited number of measurements. This can be achieved through a hash chain, where the updated value of the PCR is computed as the digest of the previous value plus the new measurement.

Authorization. This concerns how platform software can prove its authorization to call functions of the internal TPM. The TPM specification defines several types of authorizations, some or which maintain session state, to access the different entities and commands within the TPM. There are currently three authorization mechanisms:

- *Password*. A password is sent in clear with every command. No session is maintained in this authorization mechanism.
- *HMAC*. The password is set up at the beginning of the session. An HMAC is calculated on each command and responses received to determine its trustworthiness. HMAC authorizations maintain a session state.
- *Enhanced Authorization.* Are built on top of HMAC authorization sessions, and besides being based on a password, this kind of authorization also depends on TPM state including PCR values, external devices such as fingerprint readers or smart cards. Authorization conditions can be combined together to make complex authorization trees. This authorization mechanism also maintains state.

Also, the authorization mechanism defines roles, which control who can run a given command under what circumstances. There are three roles defined: user, administrator and DUP role. The DUP role is only focused to duplicate cryptographic material.

Attestation. Attestation is one of the crucial services of a TPM. It is the process by which a platform reports in a trusted way the current status of its configuration. The report can include as much information as required. The basis of the attestation are the measurements recorded in PCRs. They can then be read to know the current status of platform and be also signed to provide a secure report. The signed message can then be sent to the client. It is worth noticing that the TPM does not check the measurements, that is, it does not know whether a measurement is trustworthy or not. The trustworthiness of the measured value comes when an application uses some PCR value in an authorization policy, or remote clients ask for an attestation of some value, and later they evaluate its trustworthiness. Attestation enables such clients to confirm whether the platform has been compromised. Additionally, the TPM offers means of certifying and auditing the properties of keys and data that cross the TPM boundary.

Privacy. Every TPM has a unique public/private certified key-pair known as the Endorsement Key (EK). This can be used to prove to third parties (verifiers) that they are communicating with a genuine TPM. However, if this key is used to sign objects, it will enable verifiers to uniquely identify this TPM and link all transactions it makes. The current specification solves this issue by providing the TPM with the ability to create as many Attestation Identity Keys (AIKs) as the user wishes. AIKs act as pseudo-identity keys for the platform and can be used for different purposes and scenarios. To certify that these AIKs come from a genuine TPM, there is the need to create a new trusted entity, the Privacy Certification Authority. This approach, however, requires that the Privacy CA be highly available, as it is involved in every transaction for the creation of the AIKs. Moreover, if the Privacy CA and the verifier collude, the verifier will be able to uniquely identify a TPM.

In order to solve these issues, as well as the protocol involving the Privacy CA, there is also a protocol called **Direct Anonymous Attestation** (DAA) [8], [9], based on group signatures. This protocol does not require a highly available Privacy CA. It can transform the original credential into new unforgeable credentials that “look like” fresh credentials, while allowing that the different AIKs cannot be linked neither to the associated EK nor between them. DAA allows user-controlled linkability. That is, using the basename field of the data it provides for DAA, it can control whether a verifier can link two signatures or not.

One of the main drawbacks of the TPM privacy features comes when the EK of a TPM is compromised. Some concession on anonymity must be made to allow compromised and fraudulent TPM keys to be detected.

If, as predicted, a large-scale quantum computer becomes a reality within the next 15 years or so, existing public-key algorithms used in current TPMs will be open to attacks by this new model of computation [3].The TPM industry faces the challenge of providing a smooth transition to quantum-resistant (QR) cryptography.

Moreover, the current TPM specification came together using mostly ad-hoc progress and has grown over time. Security considerations were often considered locally during that process, but there has not been, so far, a holistic security analysis for the whole TPM specification. There was no holistic security and functional specification and design. More importantly, the security goals of the overall TPM functionality have never been formalized as a whole. This objective is independent of QR cryptography, as providing such a security model and analysis for the existing TPM is as important as it is for future TPM designs.

2.2 Relation between FutureTPM and TPM2.0

While the FutureTPM aims to contribute to the next generation (in the line) of existing TPM specifications, there will be factors to consider when researching the needs of the FutureTPM in relation to present-day TPMs. This is the case for any new generation of technology, but the addition of quantum resistance potentially adds further complexities that are worthy of attention.

2.2.1 Compromises for quantum-resistance in the TPM

There will necessarily be several compromises to make when implementing a quantum-resistant system and this is not particular to the TPM. The specific details of these will likely not be known until the systems are developed and deployed in practice, and their effectiveness assessed in light of future developments in the currently still young field of post-quantum cryptography and cryptanalysis. We can, however, outline the types of compromises that will need to be considered when designing a FutureTPM. These include:

Performance: due to the increased complexity required for quantum-hard problems, quantum-resistant algorithms will almost certainly have a direct impact on performance. This will need to be taken into account by either extending the hardware capabilities of the TPM and/or accepting less efficient functionality.

Backwards compatibility: the rift between many aspects of classical and post-quantum security means that backwards compatibility cannot be guaranteed by FutureTPM. There will be some primitives and functions that will simply no longer be able to be supported. However, this was already the case between TPM1.2 and TPM2.0 and is therefore an acceptable compromise to make in order to ensure future security. All necessary efforts will be made to achieve backwards compatibility, but we accept that some features will inevitably be deprecated. For the FutureTPM demonstrators, we will focus specifically on demonstrating the new (QR) functionalities.

Feasibility and scalability: adjacent to performance are issues of feasibility and scalability. The feasibility compromises will form a key part of this project and will be examined more thoroughly through the demonstrators. Scalability will also need to be taken into account when generalising the specific use cases and their demonstrators into a broader specification for the FutureTPM, for instance to take into account memory consumption.

Financial cost: in order to address feasibility, scalability and performance compromises, there will necessarily be a financial cost involved, although at this stage it is impossible to give a precise cost estimate as many practical challenges remain unknown. We can, however, speculate that increased

RAM and/or memory (e.g., see the previous point), as well as other hardware improvements, will incur higher production costs. There will also be associated costs with development, production and deployment, as well as the development of new, or updated, software libraries, although this is to be expected with the shift towards any new or updated platform.

Side-channel vulnerability: the TPM has historically been designed to have strong resistance to software-based attacks. However, any FutureTPM should aspire to be more secure than its predecessors, and incidents such as Rowhammer, Meltdown and Spectre [19] have shown that the line between hardware and software attacks is often blurred. The post-quantum side-channel landscape is currently underdeveloped, and there may be theoretical attacks such as superposition-based related-key attacks [6] that could fundamentally undermine security. The FutureTPM can therefore make recommendations on the need for hardware design that can mitigate the possibility of these attacks in practice.

Relation to other WPs: these issues, in particular side-channel attacks, will be covered in more detail across WP2 (for quantum security levels and the selection of new quantum resistant algorithms), WP3 (for security definitions and proofs), and WP4-5 (for risk assessment and testing in practice). The compromises outlined here are meant only as points of tension to consider when designing and implementing the FutureTPM. In any case, it is important to note that current TPMs are meant to cover software-attack only, and partially some hardware attacks for discrete TPMs (i.e., with tamper-proof capabilities), and these capabilities will also be true in future TPMs.

2.2.2 FutureTPM as a drop-in replacement for present-day TPM

With the primary goal of adding quantum resistance to existing TPM functionalities, in an ideal world the FutureTPM would be as a drop-in replacement for present-day TPMs. In some cases, this ‘plug-and-play’ approach will be achievable, although there will inevitably be many cases where this is not possible. The aim for a direct drop-in is particularly aimed at what will be outlined in Chapter 4 as the mandatory technical requirements, but there are elements of quantum resistance and other desirable features that will require adjustments. These include:

- Increased key sizes, which may exceed the memory buffer available to the hardware TPM depending on the algorithms selected in WP2;
- Key renewal and stateful processes, which may require new API calls for hash-based signatures in comparison with the lack of stateful signatures in TPM2.0;
- Key management may also provide issues for a plug-and-play approach;
- PQ keys and classical keys might not be able to co-exist in the same hierarchy;
- The current firmware updates of TPMs may not be secure under quantum security models (for example, if signed with RSA), but making firmware updates quantum resistant may require changes to the implementation from present-day TPM.

In order to aid plug-and-play potential, and for robustness in the wake of future cryptanalysis of quantum resistant primitives, the FutureTPM will utilise multiple mathematical bases for the selected primitives. This includes, for example, lattice-based [10], [11], [12], multivariate [13] and isogeny-based problems, which will be discussed in detail throughout WP2, as well as an enhancement of TPM policies for algorithm selection and adaptability in practice, which will be a focus of WP4.

2.2.3 Necessary changes to API calls

While the ‘plug-and-play’ replacement of the full list of existing TPM API calls is not easy to assess at this stage (see Section 2.2.2), we can make some useful general comments on the types of calls that may need to be adjusted. For example, in relation to the possibility of a drop-in replacement for present-day TPMs, this will mostly affect those to which the issues in Section 2.2.2 apply and this includes, in particular, key size issues and the addition of hash-based signatures that will be required for quantum resistance. The specific API that might need to be changed (for instance, DAA API), or the new API calls that will be added, will be documented in WP5.

2.2.4 Differences in security properties between FutureTPM and present-day TPM

The TPM2.0 does not commit to any security properties. Therefore, this is a weakness of present day TPMs, particularly in the case of the emerging properties of quantum security levels, but also more generally as a specification for a trusted platform. Therefore, in WP3 this project will identify the needs of security analysis and perform such analysis of the TPM system, and in WP4 this will feed into risk assessment criteria and analysis in practice. This will take into account the resistance to software-based versus hardware-based attacks mentioned in Section 2.2.1.

2.3 Consortium's Shared Vision for FutureTPM

Our vision for FutureTPM is to provide the initial research for the next generation of TPM specifications, incorporating robust and physically secure Quantum-Resistant (QR) cryptographic primitives (formally verified), to ensure long-term security, privacy and operational assurance in the complex domain of future ICT systems and services (thus, it has to be compatible with prominent computing platforms). The goal is to build on current TPM environments, based on traditional cryptography, recommending moves towards systems providing enhanced security through QR cryptographic functions, including secure authentication, encryption and signing functions, thus, turning the host device into a "hardened" security token that may also remain secure long- term against an enhanced threat landscape in quantum computing deployments. By designing an innovative portfolio of high-security QR algorithms for primitives like Key Management, Encryption, Signatures, Hash- Functions, Message Authentication Codes (MACs) and Direct Anonymous Attestation (DAA), and by taking into account a range of different types of adversaries, including remote attackers and advanced persistent threats FutureTPM will fill the perceived gaps in the current status of cybersecurity. This should occur without neglecting efficiency.

As with TPM 2.0, we envision many possible implementation scenarios for future TPMs, e.g. including discrete, integrated, firmware, software and virtual. A key strategic objective of FutureTPM is to contribute to standardization efforts at EU level within the TCG, ISO/IEC and ETSI. Because the TPM shares many functions in common with other widely-used devices, such as HSMs and TEEs, we envisage that the FutureTPM solutions will also be of benefit to the wider community of hardware-anchored security and related systems. In terms of areas of application, we expect that FutureTPM will build upon and strengthen the security of traditional applications of the TPM, for example self-encrypting drives, platform integrity measurement and strong authentication, as well as improving the security of online services, such as online banking and cloud storage.

FutureTPM should also support reactive, run-time risk assessment on all phases of the system development lifecycle, considering the complete TCG software stack. This will enable the system to generate a secure root of trust that can be used for e.g., interacting with cloud services, accessing corporate services and performing banking and eCommerce transactions. We envisage that the results of the project will lead to technology affordable by everyone, and that can be included in a very wide range of devices, such as personal computers, laptops, smartphones, Internet of Things devices and smart-cards.

Our aim is that FutureTPM will design and develop devices and algorithms that can be widely trusted, both by individuals, organizations and governments; to help ensure this ambition is realised, we will embrace an open design philosophy. We also intend that the achievements of FutureTPM will help to address current criticisms of the TPM, notably that it can be used to help "track" user activities, e.g. by manufacturers, by providing strong and provable privacy guarantees. In particular, we advocate strong visibility of who owns, and has access to, the private keys (e.g., the private endorsement key) underlying the security of the future TPM. In addition, we envision that FutureTPM will provide the basis to enhance the security functionality of systems into which they are embedded, without limiting what software a user can run on these systems. The aim is for the FutureTPM to be demonstrably applicable, usable and effective in real-world industries by adding the value of quantum resistance to the existing TPM models. The ultimate aim for the project is for the FutureTPM to make a contribution to the future specifications adopted by standardisation bodies (such as TCG or ISO) and thus be of wider benefit to the industry. Our vision is that this project will positively affect the life of everyday people by enabling them to achieve greater levels of trust in the security and privacy properties of their always-connected activities[1].

[1] In Appendix A, we detail each FutureTPM partner vision.

Chapter 3 Methodology

In this chapter, we present the methodology which has been followed for the creation of the deliverable at hand, which starts from the need to design an MVP (Minimum Viable Product) to facilitate the shared vision of the consortium, and then goes deeper to extract technical and use case requirements to formulate the critical mass of features that will synthesize the MVP

3.1 Methodology for MVP design

An MVP is the output that is able to minimize the risk of failure and improve the value generated. An MVP is a product which has the highest return on investment versus risk and is used in order to move fast towards the prototyping phase, without investing effort to features and operations which could cripple the overall development due to low utilisation an acceptance, or even due to over-complexity and design impeding factors and decisions that are not reversible. That means that many features or offerings might be tested manually, by simulating a process with human power instead of an algorithm. For example, there are some start-ups that have used humans in back office instead of AI robots, to collect information and give recommendations to users, before developing the required system which was quite costly.

Nevertheless, even if the MVP pinpoints the minimum set of features that are necessary for a product to be deployed and validated, it does not dictate a team to seize their work when reaching that state; on the contrary MVP is a strategy for cutting out unnecessary spending but being able to quickly learn about the customers and what can be sold to them, resulting in this manner into more viable, profitable and successful products that can be gradually improved and populated with extra features.

In this context, the generation of an initial MVP is essential for the smooth deployment and implementation of the project. This fact derives not only from the resources and time limit constrains for the implementation of the project but is directly related to the general FutureTPM concept that calls for the need for deliveries to follow iteration strongly looking for engagement of the all partners. As such, opting for an incremental MVP strategy seems a very justifiable and logical decision, which will set milestones and continuously test the outputs against the customer's reception.

The MVP of the FutureTPM platform, which shall include all mandatory requirements covered by the use cases, as well as the desirable that are covered by all use case, will derive out of the following procedure:

a) sort out initial technical requirements as mandatory and desirable
b) map the technical to the use case requirements in a first round of direct mapping of requirements
c) identify requirements, not directly mapped to the use cases during the first round, but which are necessary for implementation of the use cases, and conduct a second mapping round
d) conclude and order the mandatory requirements, covered by the use cases and those that are deemed as "horizontal" which will be included to the MVP
e) conclude and order the desirable requirements, that are covered by all use cases and that which will be included to the MV

This process has been performed online and also offline during the meetings of FutureTPM, with the participation of both use case and technical partners and resulted in the selection of the requirements that are more necessary to be covered during the course of the project, without of course ruling out the option to later revert decisions and turn the MVP into directions that provide more value in case any such are identified.

3.2 Requirements Elicitation Methodology

Given the clear aims of the project at the proposal stage, there was firm consensus on the essential characteristics of the FutureTPM - based on a combination of the TPM 2.0 specification and the needs of the demonstrators - and the desirable characteristics were clarified and organised into functional and security requirements.

Following this, the complete picture of the FutureTPM - both as a research topic and a practical specification - was developed with the methodology described in this section. Following this methodology made possible to map the different Technical Requirements of the Use Case requirements and design the project's MVP which describes the demonstrator of the project that will be delivered.

3.2.1 Extracting Technical Requirements

Working on the needs regarding security and privacy that derive from the current implementations of TPM and of other approach, and with a view on the project's concept, each partner submitted their narrative vision for the FutureTPM and a list of high level requirements. The lists of requirements were initially split into mandatory and desirable features, and later split further into functional and technical requirements. The first split is to acknowledge the difference between the requirements necessary to be demonstrated within the project (through the three use cases) and those requirements arising as matters of research interest or of potential use in the broader design of a FutureTPM.

Based on the underlying principle of expanding the TPM 2.0 with quantum resistance, the other requirements would be based on the project's demonstrators and any additional features that partners felt would add to the TPM's functionality and applicability for the next generation of systems. While the 'desirable' requirements might not form a core part of the project, they were included for completeness, to encourage open research questions and to lead to further independent or collaborative work by the partners which could later contribute to the impact of the project.

The second split is for clarity in dividing the requirements between functional aspects of the TPM as a device and the security it is used to enable. These were further divided into lists according to technological level, from the basic technical building blocks up to the concerns of deployment and wider adoption. The lists of requirements were then collated and formalised so that they could be systematically mapped to the requirements of the use cases to be demonstrated.

3.2.2 Extracting Use Case Requirements

Alongside the general contributions on technical requirements, the use case partners submitted a narrative of their user stories and a technical description of their intended use of the TPM, as well as which implementation(s) they would use (hardware, software, virtual). This would enable the matching of requirements to the demonstrators and leave options for further research with demonstrators being able to explore multiple implementations should time and resources allow. Use case partners worked alongside research partners to refine the requirements and expand the technical details from the narratives to specific user stories following the approach encountered in many projects using an agile approach. User stories are smaller idea segments, utilised to provide high-level definitions of requirements, describing a feature told from the end user's perspective, usually a user or customer of the system and are easier for non-technical partners to express and understand. In principle, a user story is a short, generally one-sentence, but its power is that it is time self-explanatory and that it contains enough information to describe the requirement, so that the developers can produce a reasonable estimation of the effort to implement it.

In the FutureTPM Use Cases the perspective that a User Story is really a well-expressed requirement is adopted, which [14]:

- Focuses on the viewpoint of a role who will use or be impacted by the solution
- Defines the requirement in language that has meaning for that role
- Helps to clarify the rationale for the requirement
- Helps to define high level requirements without necessarily going into low level detail too early
- Considers user goals and the business value of each requirement

Typically, user stories are proved in the following manner:

As an **< *actor role*>**, *I want* **<action to do a thing>** *so that* **<*reason*>**

In an agile project, new or updated user stories are surfacing at any time of implementation, changing the backlog. Such a behaviour is of course desired, as it helps to constantly focus on things that matter to users and exclude other features that might not be that important as the value they eventually deliver to both the system and the environment surrounding it. To ensure that user stories developed in this step have the required quality and characteristics, a user-story validation process is included into the methodology. The scope of the validation of each user story is to estimate to which extent the user story covers the INVEST [15] characteristics, i.e. the six main attributes of a good quality user story, as may be used in a Scrum [16], Kanban [17] or Extreme Programming (XP) [18] project:

- *Independent*: The user story should be self-contained in a way that there is no inherent dependency on another user story.
- *Negotiable*: User stories, up until they are part of an iteration, can always be changed and rewritten.
- *Valuable*: A user story must deliver value to the end user.
- *Estimable*: The team must always be able to estimate the size of a user story.
- *Small*: User stories should not be so big as to become impossible to plan/task/prioritize with a certain level of certainty.
- *Testable*: The user story or its related description must provide the necessary information to make test development possible.

Chapter 4 FutureTPM Use Cases

4.1 High level introduction of the FutureTPM Use Cases

During the project, three diverse Use Cases will be implemented which will be used to evaluate the FutureTPM approach in different business domains, providing not only feedback to the overall implementation, but also towards the requirements and business needs which have to be tackled by a QR TPM implementation. As such, the overall FutureTPM design will be accompanied with implementation in a hardware TPM (hTPM), software TPM (sTPM) and virtual TPM (vTPM) and will be tested through these three Use Cases. In this context, the Use Cases provide valuable input to the architectural definition of the overall FutureTPM platform (D1.2), while they also form the basis of the demonstrations and evaluation in WP6.

The different use cases will be executed as a set of scenarios and user stories, which would require a FutureTPM platform to operate as planned. In this extent, it is noted that the user stories that relate to each Use Case are not necessarily presenting an evident and explicit use of TPM, as in most cases this is a backbone operation not visible to end users; nevertheless, the successful execution of these user stories, and the unproblematic operation of each use case highly depends on the proper operation of the methods infused by FutureTPM into the Use Case.

Use Case ID	Use Case Title	TPM Implementation to be tested	Responsible Partner
1	Secure mobile wallet and payments	Hardware TPM	INDEV
2	Personal Activity and Health Kit Data Tracking	Software TPM	S5
3	Device management	Virtual TPM	HUAWEI

Table 1: The FutureTPM Use Cases

It may also be possible to explore hardware TPM in Device Management, and virtual TPM in the Activity Tracking Use Case.

It needs to be mentioned that in the three Reference scenarios presented below, a number of user stories is not directly related with TPM features or functionalities. Nevertheless, these are included in the document as they are required for the implementation of the TPM related user stories, while they also provide a better understanding of the overall operations and business value of each use case scenario. The user stories which are directly related with TPM technologies are shown in Chapter 6, under the mapping sections between the user stories and the FutureTPM technical requirements.

4.2 Reference Scenario 1 – *Secure Mobile Wallet and Payments*

4.2.1 "As-Is" Scenario

INDEV is going to use the FreePOS application as a testbed for this use case. The application allows users to charge most types of cards. The core APIs are developed using the Django Web Application Framework (Django REST). FreePOS is an actively developed app, currently ranked as one of the top finance apps in Greece, with tens of thousands of downloads. It authenticates through the user's social accounts (google or facebook) plus their phone number, which needs to be verified.

The FreePOS app is available on all platforms through a web client and additionally to iOS and Android though native mobile apps. The application connects to banking APIs to create 16-digit codes that can be redeemed in any Visa / Mastercard Card, to a user's IBAN or to a User's Viva Wallet, a mobile wallet developed by INDEV's affiliate group of companies.

The use case will revolve around the Android application due to this platform's open architecture.

As mentioned, the user must authenticate through Facebook or Google and a mobile phone number. The authentication uses an OAuth 2.0 scheme for native apps using the browser (see figure below).

OAuth is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords. This mechanism is used by companies such as Amazon, Google, Facebook, Microsoft and Twitter to permit the users to share information about their accounts with third party applications or websites.

Generally, OAuth provides to clients a "secure delegated access" to server resources on behalf of a resource owner. It specifies a process for resource owners to authorize third-party access to their server resources without sharing their credentials. Designed specifically to work with Hypertext Transfer Protocol (HTTP), OAuth essentially allows access tokens to be issued to third-party clients by an authorization server, with the approval of the resource owner. The third party then uses the access token to access the protected resources hosted by the resource server.

In the case of FreePOS, after the authentication, additional steps for actual verification are required, but we will omit those in order to stay on point with the use case requirements.

After successful authentication, the user can charge a credit card to generate a new 16-digit code or redeem an existing 16-digit code. Any actual fund transfer is handled by a PCI compliant service (Azure PaaS) that is part of the core payments infrastructure. In addition, the users can update their personal profile and view their past transactions and edit their profile details.

Most of the communications are mediated by the FreePOS service. To minimize the attack surface of the overall service, the FreePOS service does not store, send or receive sensitive data (PAN, CVV, etc). This necessitates client to server communication between the clients and the PCI compliant service. To avoid exposing the core infrastructure to the clients, authentication to these services is mediated through the FreePOS API through the use of an additional OAuth2 token.

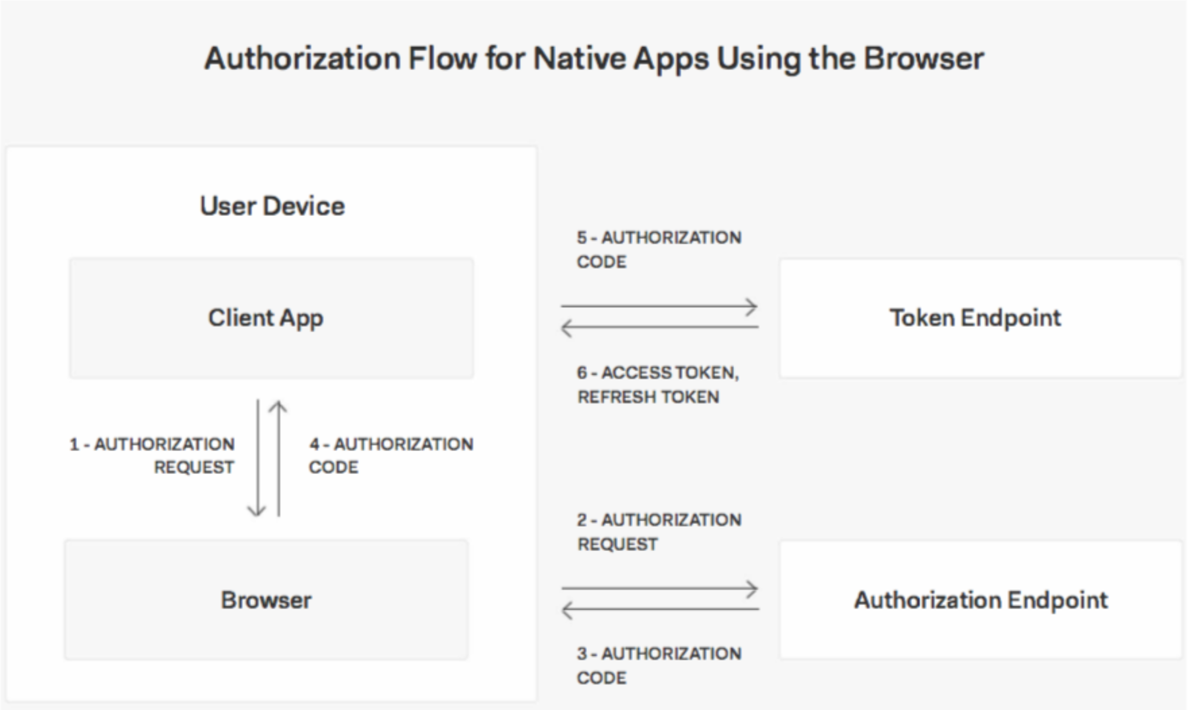
Authorization Flow for Native Apps Using the Browser
User Device
Client App
Browser
1 - AUTHORIZATION REQUEST
4 - AUTHORIZATION CODE
5 - AUTHORIZATION CODE
6 - ACCESS TOKEN, REFRESH TOKEN
Token Endpoint
2 - AUTHORIZATION REQUEST
3 - AUTHORIZATION CODE
Authorization Endpoint

Figure 1: Native App Authorization via an External User-Agent

4.2.2 Scenario’s Needs from FutureTPM

User Identification:

The current implementation of the Android app needs to store two discreet types of tokens on the device’s main storage:

- the FreePOS token that authenticates between the client and the business logic;
- the bearer token required to authenticate with the PCI compliant services.

At the present version, no TPM is used to store the above tokens. Thus, an attacker who has access to the application’s file system or memory space could steal those credentials and impersonate the user. This is actually a strong point of consideration, mainly for the following reasons:

- A lot of android users use vulnerable OS versions. This especially applies to lower end devices and devices from manufactures that don’t provide security patches to old devices.
- A lot of users chose to use unsigned software to elevate their privileges (rooting a device). Even if the rooting process is free of malware, this allows the users to run unregulated software with root privileges.

Since the application is used to conduct actual financial transactions, the above issues can be considered as highly critical.

Financial Data Confidentiality and Integrity:

The application stores the users’ past financial transactions, along with any associated metadata. The metadata includes:

- Transaction amount
- Masked PAN of the card charged (4 last digits in compliance with PCI)
- Transaction datetime
- Recipient’s phone number
- Redemption datetime
- Redemption medium

This metadata is stored on an encrypted local SQLite database. The encryption is performed via third party libraries.

Since the keys are generated and stored on the devices, the threat model is the same as above, meaning that an attacker with root privileges may gain access to the actual data.

4.2.3 “To-Be” Reference Scenario

A quantum resistant TPM can help ensure both the integrity of sensitive data and the future proofing of the mobile payments application to resist quantum attacks. INDEV plans to implement an existing **hardware TPM** 2.0 solution, provided by the appropriate project partner, to the android application. With this recommended approach, we will be able to accommodate the integration requirements of the QRTPM as it becomes available. In particular, secure computing can help in the following ways:

User Identification:

The android client is going to store all important credentials within the TPM. Those credentials include:

- OAuth Bearer tokens;
- FreePOS authentication tokens.

We are going to utilize one of the provided storage capabilities of the TPM. If the data can fit inside the TPM’s NVRAM storage, then the token is going to be stored entirely inside the TPM. Otherwise, we are going to use the module’s encryption capabilities with its self-generated secret keys and store the ciphertext externally.

Financial Data Confidentiality and Integrity:

Since the encryption is currently handled by an external library, INDEV is going to use TPM's RNG Function capability to generate the database's encryption key. As with the tokens mentioned above with the User Identification tokens, this key will be stored within the TPM.

In addition, to ensure data integrity, we are going to sign the actual encrypted database file using the TPM's HMAC (message signing) capabilities.

The above reference scenario is going to be deployed on a mirror of the actual FreePOS production infrastructure, thus providing a use case emulating real, secure mobile payments. We are planning to use an official testing API that can emulate actual monetary transactions across all media (cards, IBANs and our wallet solution).

4.2.3.1 Reference Scenario User Stories

User Stories for the Application User

INDEV.AU.1 - As an Individual User I want to log in to the FreePOS Service

User Story Confirmations:

- *User can successfully log into the platform and store the resulting token in the TPM*
- *Token type: Access Token*

INDEV.AU.2 - As an Individual User I want to tokenize my credit card

User Story Confirmations:

- *Users can authenticate against the secure PCI compliant infrastructure using tokens stored in the TPM*
- *Token type: OAuth Bearer Token*

INDEV.AU.3 - As an Individual User I want to ensure that my financial transactions are private

User Story Confirmations:

- *The local SQLite DB is encrypted with keys generated by the TPM*

INDEV.AU.4 - As an Individual User, I want to ensure that my financial transactions are not tampered with

User Story Confirmations:

- *The local SQLite DB is signed using the TPM, using HMAC (a symmetric type of signature)*

INDEV.AU.5 - As an Individual User I want to avoid revealing my credit card data to the server

User Story Confirmations:

- *Users can authenticate against the secure PCI compliant infrastructure using tokens stored in the TPM*
- *Token type: Access Token*

4.2.4 Initial Metrics of Success

1.1.1.1 Quantitative Metrics

Id	Metric	Target Value	(M)andatory / (G)ood to Have / (O)ptional
1	Credential management	100%	M
2	Key Management supported by TPM	100%	M
3	Minimize decrease in performance when utilizing TPM security operations on Android	60%	G

Table 2: Use Case #1 – Quantitative Metrics of Success

4.2.4.1 Qualitative Metrics

Id	Metric	Target Value	(M)andatory / (G)ood to Have / (O)ptional
1	Store OAuth bearer tokens in the TPM	Supported	M
2	Store Authentication tokens in the TPM	Supported	M
3	Encrypt the local database using keys generated by the TPM	Supported	M
4	Sign the local database using the TPM	Supported	M

Table 3: Use Case #1 – Qualitative Metrics of Success

4.3 Reference Scenario 2 – *Personal Activity and Health Kit Data Tracking*

4.3.1 "As-Is" Scenario

The S5Tracker second use case is based around the infrastructure build by S5 that is called S5Tracker. The S5Tracker is a cloud-based analytics engine developed by S5 acting as a data handling information environment of personalised and interlinked data streams related to activities performed mostly by individuals. The S5Tracker can be used for creating information-rich user profiles, based on activities recorded in diverse ICT communication channels and devices, pulled automatically, or inserted into the system in a semi-automatic manner by users themselves. The current information entry sources supported include APIs of specific IoT devices (e.g. Apple Health, Fitbit, Nike+, Garmin, Smart devices, etc.), Web2.0 social platforms that record users activity (such as Facebook, Twitter, etc.), as well as other smart devices that could be connected to the platform such as Smart Home kits, etc.

A strong point of the S5Tracker is the Data Anonymization and Privacy preservation service that can be used by third parties to generate aggregated "User Personas" which are fictional representative users.

As in any cloud-based data analytics engine, the development, expansion and the deployment of the service suffers from a set of systemic challenges that require continuous integration and testing efforts, as well as big time investments to undertake strategic decisions guaranteeing the service's performance and availability. In more detail, the main challenges faced at the moment, as the service resides in a public cloud provider operating as a centralised application, have to do with:

- Data sharing, privacy, confidentiality and security considerations, both at the cloud-based infrastructure as well as in the upcoming S5Tracker mobile application service;
- Data volume handling and scalability issues;
- Data processing power and system performance optimisation over the cloud-based offering.

4.3.1.1 Scenario’s Needs from FutureTPM

The current implementation of the S5Tracker is very shallow regarding both trusted computing infrastructure as well as privacy preservation methods and guarantees as the focus on the product was given mostly on data collection and analytics methods. By utilizing the infrastructure to be made available by FutureTPM, the Activity Tracking use case will be in a position to include into the overall ecosystem of its operation trusted devices. They are used at the edge of the infrastructure (e.g. at the data generation and collection points, as well as the data analysis points), which in turn will provide guarantees regarding privacy and security. These are considered highly important for the data that is being exchanged over the suggested infrastructure in order to avoid data forging incidents and data leaks, and at the same time care for privacy preservation and anonymized data delivery, while such features will be able to provide an extra layer of trust with regards to the mandates of GDPR, allowing data owners and data collectors to trust even more the entities that take part in the overall information exchange.

4.3.2 “To-Be” Reference Scenario

The use case that will be designed and developed during FutureTPM refers to the revision of the architecture of the S5Tracker infrastructure, bringing into the picture TPM methods that allow for highly trusted information exchange. In this frame, the use case has 3 main actors and 3 different components where each of those actors operates one component.

The actors identified, which play significant roles in the data value chain of the use case, and have security and privacy considerations, are the following:

- An Individual User, who is a user that collects his own data from specific sensors and social media accounts;
- A Data Analyst, who gets access to the data (anonymised data or access to personal data) to perform certain analyses;
- The S5Tracker Analytics Engine which is not an actual user but a system role that is responsible for the operation of the S5Tracker Analytics Engine.

The different components are the following:

- S5PersonalTracker - A device on the side of the “individual user” which is used primary for data collection and data push to the S5Tracker Analytics Engine;
- S5Tracker Analytics Engine – A central cloud-based service, which gets data from the S5PersonalTracker and performs some analyses online, managing individuals’ data;
- S5DataAnalysis – A computer interface used by the Data Analyst, that connects to the S5Tracker to fetch data and run online queries

As shown in the next figure, both the S5PersonalTracker and the S5DataAnalysis interfaces connect and exchange data with the S5Tracker Analytics Engine. The core focus of the use case will be to utilise software TPM methods, both at the S5PersonalTracker and at the S5DataAnalysis sides, to realise a holistic environment of privacy preservation and trust generation. In this context, privacy regarding the data owner could be achieved by enabling interconnection between the S5PersonalTracker and the S5Tracker Analytics Engine through Direct Anonymous Attestation, while at the same time, data sharing modalities towards the S5DataAnalysis side would be safeguarded, by providing access only to trusted devices for data analysis, which would be configured according to the data sharing principles of the overall platform (so that for example data cannot be exported to a storage medium). Furthermore, data coming out of the S5PersonalTracker side can be deemed as trusted and genuine, as intervention on the data collection system will be not possible as those will be signed by the TPM.

An extension to this use case may be the attempt to deem the S5Tracker Analytics Engine as trusted, thus virtual TPM will be used to allow all other interfaces to verify the integrity of that central platform. In such a scenario, both the data owners as well as the data collectors will be able to trust the platform that stores and handles the data.

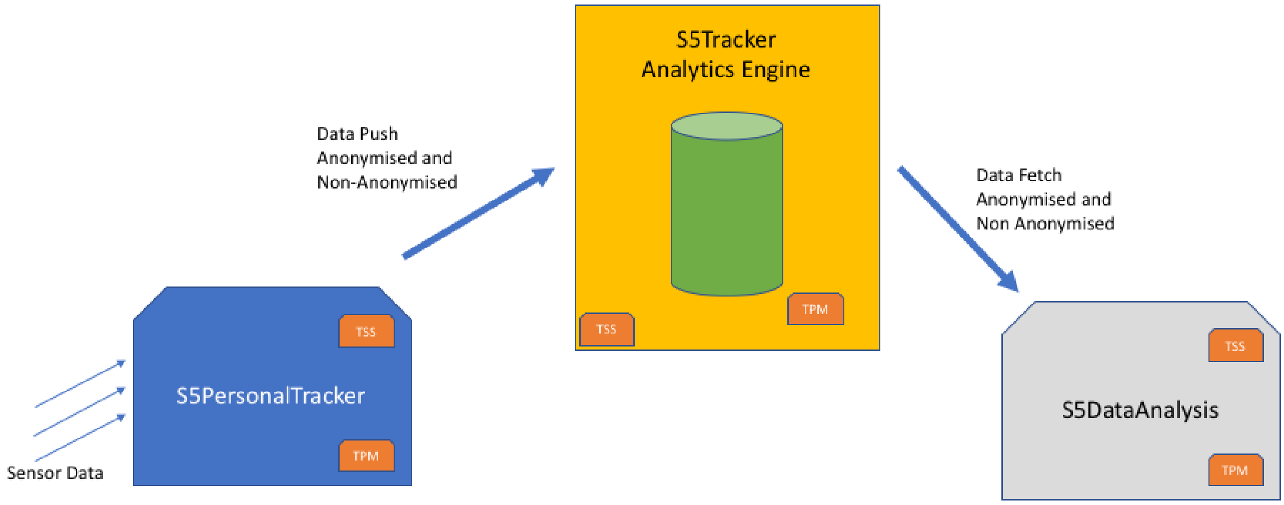

Figure 2: Personal Activity and Health kit Tracking Use Case

In the above use case, the software TPM implementation of FutureTPM will be tested. Optionally, the S5Tracker Analytics Engine can be also tested using a virtual TPM implementation

4.3.2.1 Reference Scenario User Stories

Based on the actors identified in the previous section, the user stories for the use case are split into 3 categories.

User Stories for Individual User

S5.IU.1 - As an Individual User I want to log in to the S5PersonalTracker so that I can view my profile and see my stats

User Story Confirmations:

- *User can successfully log into the platform*
- *The user can view and edit his profile data*
- *Graphs with data collected are available*
- *A user can access his account only with his email and password*
- *A user can only have 1 active session*

S5.IU.2 - As an Individual User I want to connect my activity tracking devices (e.g. Apple Health, Fitbit, etc.) to the S5PersonalTracker platform so that I can fetch data from them and store them

User Story Confirmations:

- *A user authorises the S5PersonalTracker to fetch data from activity tracking devices*
- *Data from activity tracking devices services are synced to the platform*

S5.IU.3 - As an Individual User I want to connect my social network services to the S5PersonalTracker so that I can fetch data from them and store them

User Story Confirmations:

- *A user authorises the S5PersonalTracker to fetch data from external web services*
- *Data from external web services are synced to the platform*

S5.IU.4 - As an Individual User, I want to be sure that the S5Tracker Analytics Engine I want to connect to is trusted, so that I can provide my data without issues.

User Story Confirmations:

- *S5Tracker Analytics Engine is verified as trusted through its TPM*

S5.IU.5 - As an Individual User I want to automatically register to the S5Tracker Analytics Engine through the S5PersonalTracker interface, so that I can create my online account which will be used to store the personal data upon demand

User Story Confirmations:

- *The user account is created in the S5Tracker Analytics Engine automatically upon demand of the user.*

S5.IU.6 - As an Individual User I want to upload data from the S5PersonalTracker to the S5Tracker Analytics Engine keeping my anonymity, so that I cannot be traced back through the analysis of "personas"

User Story Confirmations:

- *DAA between the S5PersonalTracker and the S5Tracker Analytics Engine*
- *S5PersonalTracker is attested but in an anonymous manner*
- *Anonymisation performed at the side of the S5PersonalTracker*
- *Anonymised data is replicated in the S5Tracker Analytics Engine*

S5.IU.7 - As an Individual User I want to provide guaranteed untampered data to the S5Tracker Analytics Engine, without disclosing my identity, so that analysts can trust that my data is genuine

User Story Confirmations:

- *S5PersonalTracker is attested but in an anonymous manner*
- *S5PersonalTracker is checked against TPM signature validity*
- *Actual data from S5PersonalTracker are stored as anonymised in the S5Tracker Analytics Engine and raw data are also encrypted in the S5Tracker Analytics Engine*

S5.IU.8 - As an Individual User I want to be able to provide access to my data (on the S5Tracker) to a specific Data Analyst, so that he could perform the analyses he wants

User Story Confirmations:

- *The S5DataAnalysis side gets a private key for decrypting an Individual Users data*
- *S5DataAnalysis side gets access to a specific dataset from an Individual User*

S5.IU.9 - As an Individual User I want to be able to remove my personal, non-anonymised data from the S5Tracker Analytics Engine, so that no one else can have access to those.

User Story Confirmations:

- *All persona, non-anonymized data owned by an Individual User are removed from the S5TrackerPlatform*

User Stories for the Data Analyst

S5. DA.1 – As a Data Analyst, I want to be able to have access to the data of individuals without being able to transfer them, so that I can be trusted to perform my analyses in a privacy preserving manner

User Story Confirmations:

- *The Data Analyst device is checked by the S5Tracer Analytics Engine against his TPM signature validity*
- *The Data Analyst is able to log in to the S5Tracker Analytics Engine only with its own email and password*

- *The S5DataAnalysis interface can fetch anonymised, obfuscated data (which are processed by the S5Tracker Analytics Engine to remove personal information and perform data obfuscation to minimise possibilities for back-tracing) from individual users stored on the S5Tracker Analytics Engine*
- *A S5DataAnalysis interface can only have 1 active session*

S5. DA.2 - As a Data Analyst, I want to be sure that the S5Tracker Analytics Engine I want to connect to is trusted, so that I can get data without issues.

User Story Confirmations:

- *S5Tracker Analytics Engine is attested as trusted through its TPM*

S5. DA.3 - As a Data Analyst I want to register to the S5Tracker Analytics Engine so that I can create my data analyst online profile

User Story Confirmations:

- *A Data Analyst profile is available on the S5Tracker Analytics Engine*

S5. DA.4 - As Data Analyst I want to connect to the S5Tracker Analytics Engine to view specific data related to an individual user once he has provided me access, so that I can better understand his activity

User Story Confirmations:

- *The S5DataAnalysis side is verified by the S5Tracker Analytics Engine as trusted through its TPM*
- *The S5DataAnalysis has access to personal data of an Individual User*
- *The S5DataAnalysis obtains a private key by the S5Tracker Analytics Platform to decrypt the data*

User Stories for the S5Tracker Analytics Engine

S5.AE.1 - As the S5Tracker Analytics Engine I want to store anonymised data coming from Individual Users, so I can use them later for serving analytics

User Story Confirmations:

- *Anonymised data coming from Individual Users are available on the S5Tracker Analytics Engine*

S5.AE.2 - As the S5Tracker Analytics Engine I want to store non-anonymised data coming from Individual Users, so I can use them later for serving analytics

User Story Confirmations:

- *Non-Anonymised data coming from Individual Users are available on the S5Tracker Analytics Engine*

S5.AE.3 - As the S5Tracker Analytics Engine I want to acknowledge that a device used by an Individual User is trusted, so I can allow information exchange

User Story Confirmations:

- *The S5PersonalTracker side is attested by the S5Tracker Analytics Engine*

S5.AE.4 - As the S5Tracker Analytics Engine I want to acknowledge that a device used by a Data Analyst is trusted, so I can allow information exchange for non-obfuscated data coming from an Individual User.

User Story Confirmations:

- *The S5Data Analyst's side is attested by the S5Tracker Analytics Engine*

S5.AE.5 - As the S5Tracker Analytics Engine I want to prove that as a platform I have not been compromised regarding my initial configuration, so that I am trusted by other entities

- *The S5Tracker Analytics Engine s attested by other partners*

4.3.3 Initial Metrics of Success

4.3.3.1 Quantitative Metrics

Id	Metric	Target Value	(M)andatory / (G)ood to Have / (O)ptional
1	Prevention of S5Tracker Analytics Engine mimicking by other platforms via TPM usage	100%	O
2	Successful attempts at breaching confidentiality / gaining unauthorised access to personal data	0	O
3	Improved perception of Individual Users' trust to S5PersonalTracker as a data hub	100%	G
4	Improved perception of Individual Users' trust to third parties handling their data	50%	G

Table 4: Use Case #2 – Quantitative Metrics of Success

4.3.3.2 Qualitative Metrics

Id	Metric	Target Value	(M)andatory / (G)ood to Have / (O)ptional
1	Improve anonymity of S5PersonalTracker interfaces	Supported	M
2	Trust guarantees for the S5DataAnalysis interface	Supported	M
3	DAA for the S5PersonalTracker	Supported	M
4	Trust guarantees for the S5Tracker Analytics Engine	Supported	O

Table 5: Use Case #2 – Qualitative Metrics of Success

4.4 Reference Scenario 3 – *Device Management*

4.4.1 "As-Is" Scenario

The device management use case focuses on the management of network infrastructures, such as those of enterprise companies. Companies often define strict security policies to protect their valuable data and rely on the network infrastructure to support them. However, the infrastructure will not behave as expected if a network device is compromised. A tampered device could for example leak the data to a location chosen by the attacker or expose it for brute-forcing. The management of the network infrastructure should take into account the integrity of the managed devices, for example with techniques introduced by trusted computing.

Initially, the focus of the demonstrator work will be on defining a network infrastructure that can be used as a reference for evaluating the benefits introduced by trusted computing. The enhancements that will be developed are intended to be applied on bigger and more complex infrastructure. The figure below illustrates an initial version of the infrastructure that can be used as a basis:

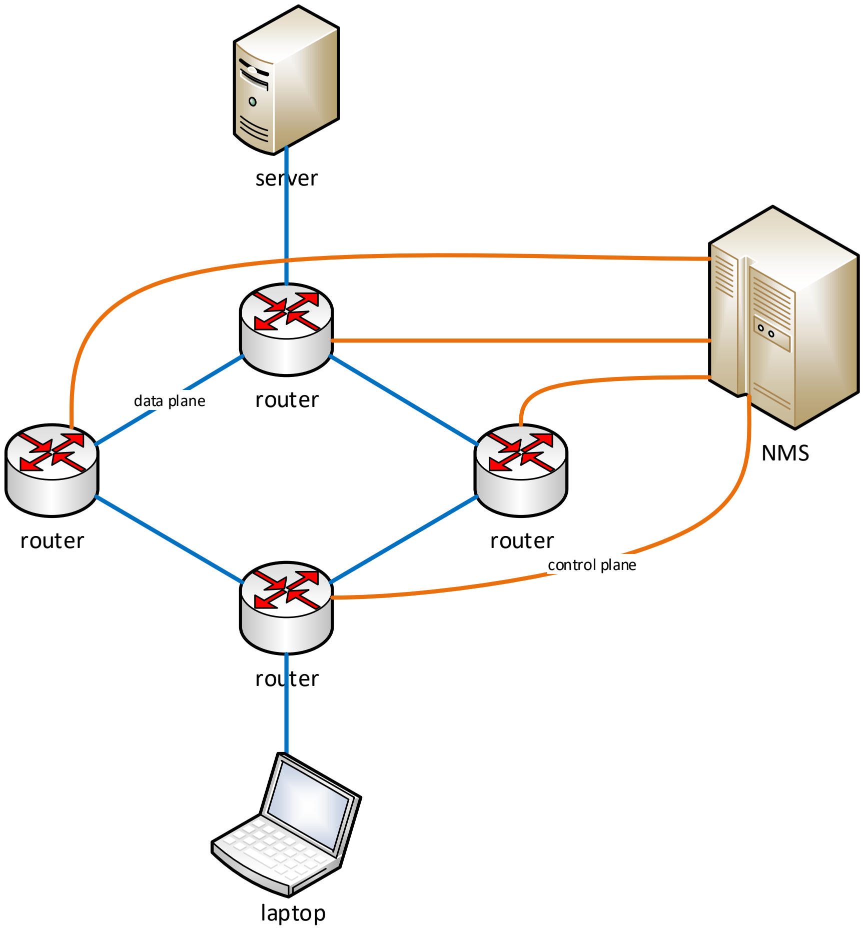

Figure 3: Network Infrastructure

The figure shows an enterprise network, composed of four routers, which interconnects a laptop and a server. Network packets can take alternative paths and the infrastructure usually selects the path depending on the workload on the routers.

The laptop retrieves confidential data from the server through an end-to-end secure channel (e.g. TLS), so that network devices could not read or modify the data in transit. Although the communication is protected, network devices could still leak metadata about the traffic or can expose it for subsequent attacks on the secure protocol and/or cryptographic algorithms. To reduce such risk, the routing policy in the network could be made to depend also on the system integrity of the router and not just on its workload. A compromised router would be avoided by having the traffic redirected through uncompromised routers. A trade-off between the risk of attack on the traffic and the availability of connectivity could be made in peak traffic scenarios.

Network devices are centrally managed by a Network Management System (NMS). This component has a global view of the network and can determine and send the configuration to each device. Management commands are sent through dedicated channels part of the *control plane*, while packets exchanged between the laptop and the server are sent through the *data plane*.

A sample management operation on an infrastructure without trusted computing capability is depicted in the figure below:

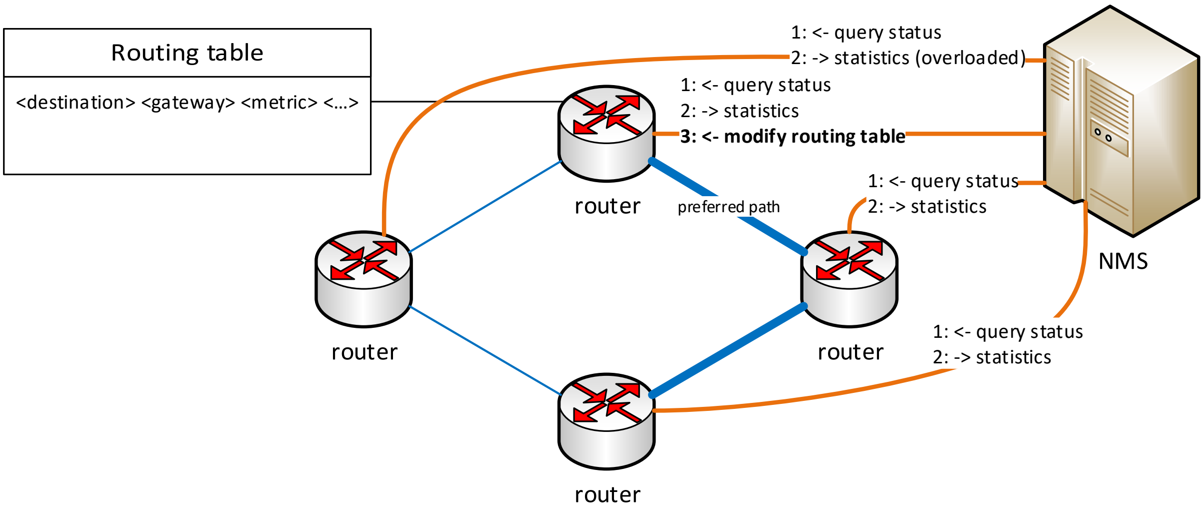

Figure 4: Sample Device Management Scenario

The NMS queries each router to obtain information about their workload. If a router is overloaded, the NMS tries to distribute the traffic to less busy routers[2]. The NMS does not take into account the integrity status of the routers to decide which routing tables should be modified.

The second part of the work on the demonstrator will consist in adding trusted computing capability to routers, to address the issues described in the next section.

4.4.1.1 Scenario's Needs from FutureTPM

Without trusted computing or similar technologies, routers face the following challenges:

Device identification: identification is usually done by establishing a secure channel between the NMS and the device, where the device must prove the possession of a key; however, the key is not strongly bound to the device and may be leaked or duplicated to other devices.

Software integrity: the NMS cannot determine if the management commands sent to the controlled devices have been processed successfully and does not have trustworthy evidence on the integrity of the device. A compromised device could e.g. report that is not overloaded, so that the NMS enables it to route traffic.

Data integrity and confidentiality: data is often stored in plain text and integrity is not verified; also, data can be accessed by the device even when compromised.

4.4.2 "To-Be" Reference Scenario

Trusted computing significantly helps to solve the challenges above. The TPM contains a unique key that is never revealed outside the chip in plain text and can be used for device identification. The TPM also includes Platform Configuration Registers (PCRs), which can be used to accumulate measurements of the software executed in the platform and provide them in a trustworthy way (i.e. TPM quote). With measurements, the NMS can determine if the software on controlled devices are correctly enforcing the management commands. Data integrity and confidentiality can be achieved by using TPM keys. It is possible to define policies to allow the key usage depending on the integrity of the software. The updated flow is depicted in the figure below.

[2] This is in addition to dynamic routing protocols running among routers without NMS intervention. Such protocols are out-of-scope for the demonstrator.

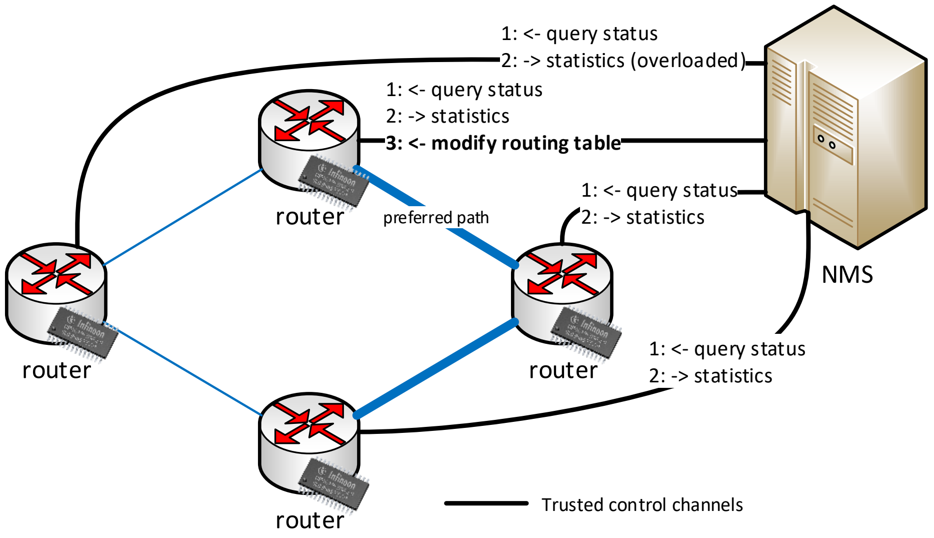

Figure 5: Device Management Scenario with Trusted Computing

Before the NMS exchanges management information with a router, it establishes with it a trusted channel. Trusted channels are secure channels (like e.g. TLS) for which the integrity of the endpoints is verified during the setup phase and periodically during communication. The enhanced NMS will make security decisions depending on whether trusted channels can be established and depending on the policy defined by the Network Administrator. A policy could require the NMS to divert the traffic away from a compromised router, as the same as if that router was overloaded.

Routers are also enhanced to use TPM keys. The NMS will ensure that the TPM keys are bound not only to the device but also to an uncompromised operating system. If a router is compromised, the TPM denies the usage of that key and the trusted channel with the NMS cannot be established.

4.4.2.1 Reference Scenario User Stories

As described in Figure 1, the network infrastructure consists of three components: NMS, router and data plane endpoints (laptop and server). The NMS and, through it, the router are operated by the Network Administrator, who is inherently trusted for configuring the routers. The router is physically installed and connected by a Network Operator, which is untrusted (i.e. has no role in establishing trust relationships and does not configure anything). The laptop is operated and/or managed by the End User.

User Stories for the Network Administrator

HWDU.NA.1 – As a Network Administrator, I want to enrol the router with the NMS so that it is accepted in the network infrastructure

User Story Confirmations:

- *The router appears in the list of devices managed by the NMS based on its TPM-based identity*

HWDU.NA.2 – As a Network Administrator I want to define a trusted routing policy on the NMS so that the traffic is processed according to the trust states of routers

User Story Confirmations:

- *A routing policy depending, among others, on the trust state of routers is defined in the NMS*

HWDU.NA.3 – As a Network Administrator I want to enforce the trusted routing policy in the network to reduce the risk of traffic leaking by untrusted routers

User Story Confirmations:

- *Routing tables on adjacent routers are modified when the trust state of a given neighbouring router changes*

HWDU.NA.4 – As a Network Administrator I want to monitor the overall trust state of the network infrastructure

User Story Confirmations:

- *The NMS displays the trust state and routing table for each router in the network*

User Stories for the Network Operator

HWDU.NO.1 – The Network Operator connects the router to the network

User Story Confirmations:

- *A TPM key is generated on the router for use to establish trusted channels*
- *The TPM key is validated by the NMS*
- *A trusted management channel is established between the NMS and the router*
- *An LED light on the router case indicates that the router has connected to the NMS*

User Stories for the End User

HWDU.EU.1 – As an End User, I want to access a web application hosted in a server that is remotely connected via the network of routers managed by the NMS

User Story Confirmations:

- *The End User's browser successfully connects to the server and displays the application*

Alternative User Story Confirmation (server unreachable, potentially due to routing policy disallowing untrusted routers):

- *The End User's browser cannot connect to the web server*

4.4.3 Initial Metrics of Success

4.4.3.1 Quantitative Metrics

Id	Metric	Target Value	(M)andatory / (G)ood to Have / (O)ptional
1	Amount of routers whose integrity is monitored by NMS	100%	M
2	Amount of routers hiding their integrity status	0%	M
3	Amount of detected integrity attacks on routers	80% (with integrity models) 60% (standard IMA)	M
4	Amount of traffic diverted to alternative paths when a router is compromised	80%	G
5	Amount of files whose integrity can be verified	100% (with integrity models) 99% (standard IMA)	G M

Table 6: Use Case #3 – Quantitative Metrics of Success

4.4.3.2 Qualitative Metrics

Id	Metric	Target Value	(M)andatory / (G)ood to Have / (O)ptional
1	Traffic routing based on router trust state	Supported	M
2	Trusted channels between NMS and each router in the network	Supported	M
3	Device authentication key for trusted channel protected by TPM	Supported	M
4	Integrity protection of router configuration data using a TPM key	Supported	M

Table 7: Use Case #3 – Qualitative Metrics of Success

Chapter 5 FutureTPM Technical Requirements

In this chapter, we describe the technical requirements of FutureTPM, which have been clustered in mandatory and desirable ones. This split differentiates the requirements that are needed for the demonstrators within the FutureTPM project itself, and the possible requirements of a FutureTPM as a standard in general (and as a topic of research). *Thus, it is the mandatory requirements that will form the basis of the core technical requirements of this project.*

5.1 Technical Requirements

In the following, we gather the list of technical requirements. We have split the list of technical requirements in mandatory requirements (see Sect. 5.1.1) and in desirable requirements (see Sect. 5.1.2).

5.1.1 Mandatory Technical Requirements

We have clustered the list of technical requirements in the following classes: basic blocks, performance and cost-effectiveness, implementation and deployment, and third-party support/compatibility and standardization. These groupings are by type of requirements based on the level of design, from core cryptographic principles up to overarching aims for widespread adoption and deployment.

5.1.1.1 Basic Blocks

The set of basic functionalities that must be provided by future TPMs are as follows. These are the functional basis of existing TPMs, and therefore play an essential role in defining any future versions of the TPM.

A FutureTPM must provide the same functionalities of TPM 2.0 [20], namely:

- *[TR.1.1.1]* It should provide non-volatile random-access memory (NVRAM) storage;
- *[TR.1.1.2]* It should provide a small set of platform configuration registers (PCR);
- *[TR.1.1.3]* It should support protocol and algorithm agility, i.e. the new TPM must allow application developers to choose among various cryptographic primitives and protocols that provide a specific service;
- *[TR.1.1.4]* It should support enhanced authorization (EA).

5.1.1.2 Performance and cost-effectiveness

Future TPMs have to also consider the impact on performance and the additional cost of implementation (e.g., chip area), in particular the cryptographic primitives and protocols implemented in the new TPM must be efficient enough to be used in practice. In order for TPMs to support a range of devices and implementations (particularly hardware for smaller embedded devices such as those used in the Internet of Things or infrastructure), efficiency is a core concern. This will also have an economic impact on, for example, manufacturing costs. Nevertheless, security should not be compromised, so the specific algorithms chosen for the FutureTPM must balance security and functionality. Specifically:

- *[TR.1.2.1]* It should be feasible to implement the chosen post-quantum algorithms on platforms with restricted memory, while providing an acceptable performance;
- *[TR.1.2.2]* The selected crypto algorithms can be implemented securely on an identified platform, e.g.:

  - x86 for firmware TPM, relatively constrained 32-bit CPUs, smartcards, FPGA/ASIC (for full HW-accelerated implementations).

5.1.1.3 Implementation and deployment

This set of requirements considers the impact of future TPM in real use cases. Primarily, these are practical considerations for the three specific demonstrators to be used in this project, including implementations of the TPM as hardware, software and virtualised platforms. More broadly and building on the lessons learned in the demonstrators through this research, the requirements for practical deployment should be considered when designing a specification for the FutureTPM that can be easily adopted by industry for real world situations.

- *[TR.1.3.1]* Selected algorithms should be chosen in such a way that it is possible to enhance the performance of the cryptographic calculations with a small hardware coprocessor, in particular:
  - as existing hardware coprocessors can be reused for lattice-based cryptography, the TPM should at least implement one or more lattice-based algorithms.
- *[TR.1.3.2]* Selected post-quantum cryptographic primitives should be chosen to allow for reuse of hardware accelerator engines, for example,
  - an accelerator for lattice-based cryptography may be used for several cryptographic operations, or an accelerator for hashing may also support hash-based signatures.
- *[TR.1.3.3]* Development and testing of a software FutureTPM, including adequate support for the Trusted Software Stack (TSS);
- *[TR.1.3.4]* Development and testing of a virtual FutureTPM, including adequate support for the Trusted Software Stack (TSS);
- *[TR.1.3.5]* Development and testing of a hardware FutureTPM (evaluation board), including adequate support for the Trusted Software Stack (TSS);
- *[TR.1.3.6]* Allow support for some legacy primitives/protocols.

5.1.1.4 Third party support/compatibility and standardization

Future TPMs should also consider the integration with existing products, in particular operating systems (OSes), and support for standardization in international bodies. The TPM is not and should not be platform-specific. Building on the reputation of the existing TPM as a set of standards with multiple implementations, the FutureTPM should also be able to be used by a wide range of manufacturers, administrators, developers and end users. As with previous versions of the TPM, it should also aim to be supported by international organisations (such as the Trusted Computing Group).

- *[TR.1.4.1]* Provide support for at least one major OS (e.g. Linux).

5.1.2 Desirable Technical Requirements

We have clustered the desirable technical requirements in the following classes: performance, implementation, and third-party support/compatibility.

5.1.2.1 Performance

The impact of FutureTPM on performance should be minimal, hence:

- *[TR.2.1.1]* The efficiency of FutureTPM primitives and protocols should be similar or better than the ones currently provided by TPM 2.0;

- *[TR.2.1.2]* The use of FutureTPM to attest (during run-time) code snippets running in an embedded system should be similar or better than the current ones (also benchmarked against the use of Trusted Execution Environments).

5.1.2.2 Implementation

There are few desirable requirements that have an impact on implementation, in particular:

- *[TR.2.2.1]* The cryptographic primitives should provide the basis for a flexible implementation of cryptographic algorithms and protocols:
  - for example, they allow a design with small and reusable hardware accelerators that can be used for multiple cryptographic algorithms;
  - only changes in software are required to adapt the implementation to support other algorithms and protocols;
- *[TR.2.2.2]* The implementation of easily extendable software and hardware TPM systems may facilitate research in the future;
- *[TR.2.2.3]* Trying to integrate existing APIs with any type of TPM (QR or otherwise);
- *[TR.2.2.4]* Developing a Python based API or Library that works with Django REST Framework and integrates the TPM developed;
- *[TR.2.2.5]* Enhancing the architecture of virtual TPM to provide better security guarantees closer to that of a physical TPM;
- *[TR.2.2.6]* Some ability to run arbitrary code, e.g. in a sandboxed environment (similar to Javacards).

5.1.2.3 Third party support/compatibility

Finally, there are few other requirements that have to deal with third party support and compatibility that are however desirable, such as:

- *[TR.2.3.1]* The TPM should be virtualization-aware:
  - e.g., to facilitate context-switch among VMs for key registers.
- *[TR.2.3.2]* Easy to support on mobile and IoT devices;
- *[TR.2.3.3]* Easy to port to existing architectures;
- *[TR.2.3.4]* Long-life time (e.g., easy to update in IoT context).

5.2 Security Requirements

In the following, we gather the list of security requirements. We have split the list of security requirements in mandatory requirements (see Sect. 5.2.1) and in desirable requirements (see Sect. 5.2.2).

5.2.1 Mandatory Security Requirements

We cluster the mandatory security requirements for future TPMs in the following classes: primitives, PQ security, integrity requirements, and data privacy.

5.2.1.1 Primitives

FutureTPMs must provide all services offered by TPM 2.0 through QR cryptographic functions including secure authentication, encryption and signing functions. These primitives form the basic security functionality that supports more complex operations. It is therefore essential to include both symmetric cryptography (used for tasks such as encrypted offloading of TPM memory into a harddrive) and asymmetric cryptography (also known as public key cryptography and used for establishing secure connections such as connecting securely with a website). Namely, this includes:

- *[SR.1.1.1]* Pseudorandom number generator;
- *[SR.1.1.2]* Key generation and storage functionalities;
- *[SR.1.1.3]* Hash functions;
- *[SR.1.1.4]* MAC;
- *[SR.1.1.5]* Symmetric encryption;
- *[SR.1.1.6]* Digital signatures;
- *[SR.1.1.7]* Public key encryption and key exchange;
- *[SR.1.1.8]* Direct Anonymous Attestation (DAA) [for SW TPM].

5.2.1.2 PQ security

The support for QR primitives is one the main cornerstones of FutureTPM. To future-proof the TPM in the context of quantum computers which will break many aspects of conventional cryptography, the primitives listed above must be resistant to attack by quantum computers.

- *[SR.1.2.1]* Support for possible QR-crypto candidates for each category (symmetric, asymmetric and DAA);
- *[SR.1.2.2]* QR Support for signing, key exchange, attestation;
- *[SR.1.2.3]* Reach QS-Level 1 (post-quantum crypto);
- *[SR.1.2.4]* Provide a crypto library with TPM backed keys implementing TLS with QR algorithms.

5.2.1.3 Integrity requirements

One of the main functionalities of TPMs is related to software integrity. This includes verifying that the software running on a device is trustworthy and has not been tampered with by intruders or malware. Therefore, FutureTPM must offer the same functionality of TPM 2.0:

- *[SR.1.3.1]* Support *s*oftware measurement (PCR extend) and measurement reporting (Quote), using QR algorithms;
- *[SR.1.3.2]* Support remote attestation functionalities;
- *[SR.1.3.3]* Support sealing and binding operations.

5.2.1.4 Data privacy

One key aspect of future TPM is the privacy guarantees of the data stored. At base, this is the goal of any cyber security system, ensuring your data is protected from intruders, which the TPM seeks to enable even if the device as a whole may be compromised.

- *[SR.1.4.1]* Allow the protection of sensitive information;
- *[SR.1.4.2]* It should be hard for an adversary to learn the secret information required for any action (e.g., authentication, encryption, etc.);
- *[SR.1.4.3]* Credentials should be stored on user device and must be protected from eavesdropping/leakage.

5.2.2 Desirable Security Requirements

We have clustered the list of desirable security requirements in the following classes: PQ security, and support for additional application areas.

5.2.2.1 PQ security

PQ security primitives can be extended to include further functionalities.

- *[SR.2.1.1]* The future TPM implements a PQ-DAA scheme;
- *[SR.2.1.2]* Reach QS-Level 2 (superposition-based quantum security):
  - Quantum security level 2 (quantum adversary and interaction, pq crypto and side-channel mitigation required);
- *[SR.2.1.3]* More than two possible QR-crypto candidates for symmetric and asymmetric primitives;
- *[SR.2.1.4]* Support for a PFS end-to-end encryption protocol, e.g. a PQ variant of Signal's Double Ratchet algorithm, or at least provide the (combination of) primitives the TPM would need to support secure implementation of such an algorithm;
- *[SR.2.1.5]* Add support for QR algorithms and for TPM as key storage back-end to an IPSEC (IKE) implementation;
- *[SR.2.1.6]* The new TPM should implement cryptographic primitives and protocols that are secure against adversaries with quantum capabilities:
  - If for certain services, available quantum-resistant primitives or protocols are not suitable (e.g., for efficiency reasons), the new TPM may implement cryptographic primitives and protocols that are partially vulnerable to adversaries with quantum capabilities (e.g., protocols where some security properties hold against quantum adversaries, but others do not);
- *[SR.2.1.7]* Provide resiliency capability to the TPM internal architecture to enable provable recovery in case of TPM firmware vulnerability or QR algorithm compromise.

5.2.2.2 Support for additional application areas

Finally, we list some desirable functionalities for emerging application areas that future TPMs may wish to support. These areas include, but are not limited to, remote voting, anonymous communications, virtual private networks, cryptocurrency-related scenarios, cloud computing and IoT.

- *[SR.2.2.1]* Support for a broader range of access policies;
- *[SR.2.2.2]* Functionality for key translation (i.e, re-encrypting a given message under a different key);
- *[SR.2.2.3]* Use of FutureTPM to support cryptographic operations in blockchains and other services such as verifiable data access:
  - E.g., if a user that has access to the data in a block did actually use the data.
- *[SR.2.2.4]* Support for mixnet functionality (for privacy-related features, e.g. contact discovery as it is being implemented in Signal);
- *[SR.2.2.5]* Secure logging of access to cryptographic operations in a blockchain (ability to provide accountable decryption and similar constructs);
- *[SR.2.2.6]* Secure key backup and recovery on TPM damage.

Chapter 6 Definition of the FutureTPM MVP

6.1 Mapping of Use Case Requirement to FutureTPM Technical Requirements

In this chapter, we provide tables that map the use case requirements from the user stories to the requirements on the FutureTPM. The outcome is that the FutureTPM is required to help secure computations and connections in a variety of circumstances. There are two ways this could be achieved on the FutureTPM:

- [Most secure but inflexible] The TPM API is expanded to include functionality specific to existing protocols, such as TLS and OAuth.
- [Less secure but more flexible] The TPM API is not expanded. The TPM is used to measure and attest platform software which has the functionality specific to those protocols and the use cases.

For each reference scenario there is the possibility that some user stories do not need the interaction with a TPM. See for example HWDU.EU.1 from the Device Management reference scenario, where an end user wishes to access a remote web server. Consequently, the corresponding columns on the mappings tables are empty, as there is no technical requirement from FutureTPM that can be showcased in that user story.

The tables follow the coding convention from Chapter 3 for user stories (first row), and from Chapter 5 for technical requirements (first column).

6.1.1 Secure Mobile Wallet and Payments

Requirement ID	INDEV. AU.1	INDEV. AU.2	INDEV. AU.3	INDEV. AU.4	INDEV. AU.5
Mandatory Technical Requirements					
TR.1.1.1	x	x			x
TR.1.1.2					
TR.1.1.4	x	x	x	x	x
TR.1.2.1					
TR.1.2.2	x	x	x	x	x
TR.1.3.1					
TR.1.3.2					
TR.1.3.3					
TR.1.3.4					
TR.1.3.5	x	x	x	x	x
TR.1.3.6					
TR.1.4.1	x	x	x	x	x
Desirable Technical Requirements					
TR.2.1.1	x	x	x	x	x
TR.2.1.2					
TR.2.2.1					
TR.2.2.2					
TR.2.2.3	x	x	x	x	x
TR.2.2.4	x	x	x	x	x
TR.2.2.5					
TR.2.2.6					
TR.2.3.1					
TR.2.3.2	x	x	x	x	x

Requirement ID	INDEV. AU.1	INDEV. AU.2	INDEV. AU.3	INDEV. AU.4	INDEV. AU.5
TR.2.3.3	x	x	x	x	x
TR.2.3.4					
Mandatory Security Requirements					
SR.1.1.1					
SR.1.1.2	x	x	x		x
SR.1.1.3				x	
SR.1.1.4				x	
SR.1.1.5			x		
SR.1.1.6				x	
SR.1.1.7					
SR.1.1.8					
SR.1.2.1	x	x	x	x	x
SR.1.2.2				x	
SR.1.2.4					
SR.1.3.1					
SR.1.3.2					
SR.1.3.3					
Desirable Security Requirements					
SR.2.1.1					
SR.2.1.2					
SR.2.1.3					
SR.2.1.4					
SR.2.1.5					
SR.2.1.6					
SR.2.1.7					
SR.2.2.1	x	x	x	x	x
SR.2.2.2					
SR.2.2.3					
SR.2.2.4					
SR.2.2.5					
SR.2.2.6	x	x	x	x	x

Table 8: Use Case #1 – Mapping Use Case Requirements to Technical Requirement

6.1.2 Personal Activity and Healthkit Tracking Use Case

Requirement ID	S5. IU.1	S5. IU.2	S5. IU.3	S5. IU.4	S5. IU.5	S5. IU.6	S5. IU.7	S5. IU.8	S5. IU.9	S5. DA.1	S5. DA.2	S5. DA.3	S5. DA.4	S5. AE.1	S5. AE.2	S5. AE.3	S5. AE.4	S5. AE.5
Mandatory Technical Requirements																		
TR.1.1.1																		
TR.1.1.2				x		x	x			x	x		x	x		x	x	x
TR.1.1.4				x		x	x			x	x		x	x		x	x	x
TR.1.2.1						x	x											
TR.1.2.2																		
TR.1.3.1																		
TR.1.3.2																		
TR.1.3.3				x		x	x			x	x		x	x		x	x	x
TR.1.3.4																		
TR.1.3.5																		
TR.1.3.6																		
TR.1.4.1				x		x	x			x	x		x	x		x	x	x
Desirable Technical Requirements																		
TR.2.1.1				x		x	x			x	x		x	x		x	x	x
TR.2.1.2																		
TR.2.2.1																		
TR.2.2.2																		

Requirement ID	S5. IU.1	S5. IU.2	S5. IU.3	S5. IU.4	S5. IU.5	S5. IU.6	S5. IU.7	S5. IU.8	S5. IU.9	S5. DA.1	S5. DA.2	S5. DA.3	S5. DA.4	S5. AE.1	S5. AE.2	S5. AE.3	S5. AE.4	S5. AE.5
TR.2.2.3																		
TR.2.2.4				x		x	x			x	x		x	x		x	x	x
TR.2.2.5																		
TR.2.2.6																		
TR.2.3.1																		
TR.2.3.2																		
TR.2.3.3				x		x	x			x	x		x	x		x	x	x
TR.2.3.4																		
Mandatory Security Requirements																		
SR.1.1.1				x		x	x			x	x		x	x		x	x	x
SR.1.1.2				x		x	x			x	x		x	x		x	x	x
SR.1.1.3				x		x	x			x	x		x	x		x	x	x
SR.1.1.4				x		x	x			x	x		x	x		x	x	x
SR.1.1.5				x		x	x			x	x		x	x		x	x	x
SR.1.1.6				x		x	x			x	x		x	x		x	x	x
SR.1.1.7													x					
SR.1.1.8						x								x				
SR.1.2.1				x		x	x			x	x		x	x		x	x	x
SR.1.2.2				x		x	x			x	x		x	x		x	x	x
SR.1.2.4																		
SR.1.3.1				x		x	x			x	x		x	x		x	x	x
SR.1.3.2				x		x	x			x	x		x	x		x	x	x
SR.1.3.3																		
Desirable Security Requirements																		
SR.2.1.1																		
SR.2.1.2																		
SR.2.1.3																		
SR.2.1.4																		
SR.2.1.5																		
SR.2.1.6																		
SR.2.1.7																		
SR.2.2.1																		
SR.2.2.2																		
SR.2.2.3				x		x	x			x	x		x	x		x	x	x
SR.2.2.4																		
SR.2.2.5				x		x	x			x	x		x	x		x	x	x
SR.2.2.6																		

Table 9: Use Case #2 – Mapping Use Case Requirements to Technical Requirement

6.1.3 *Device Management Use Case*

Requirement ID	HWDU. NA.1	HWDU. NA.2	HWDU. NA.3	HWDU. NA.4	HWDU. NO.1	HWDU. EU.1
Mandatory Technical Requirements						
TR.1.1.1	x			x	x	
TR.1.1.2	x			x	x	
TR.1.1.4	x			x	x	
TR.1.2.1	x			x	x	
TR.1.2.2						
TR.1.3.1						
TR.1.3.2						
TR.1.3.3						
TR.1.3.4	x			x	x	
TR.1.3.5						
TR.1.3.6	x			x	x	

Requirement ID	HWDU.NA.1	HWDU.NA.2	HWDU.NA.3	HWDU.NA.4	HWDU.NO.1	HWDU.EU.1
TR.1.4.1	x			x	x	
Desirable Technical Requirements						
TR.2.1.1	x			x	x	
TR.2.1.2	x			x	x	
TR.2.2.1	x			x	x	
TR.2.2.2						
TR.2.2.3	x			x	x	
TR.2.2.4						
TR.2.2.5	x			x	x	
TR.2.2.6						
TR.2.3.1						
TR.2.3.2						
TR.2.3.3	x			x	x	
TR.2.3.4	x			x	x	
Mandatory Security Requirements						
SR.1.1.1	x			x	x	
SR.1.1.2	x			x	x	
SR.1.1.3	x			x	x	
SR.1.1.4	x			x	x	
SR.1.1.5	x			x	x	
SR.1.1.6	x			x	x	
SR.1.1.7	x			x	x	
SR.1.1.8						
SR.1.2.1	x			x	x	
SR.1.2.2	x			x	x	
SR.1.2.4	x			x	x	
SR.1.3.1	x			x	x	
SR.1.3.2	x			x	x	
SR.1.3.3	x			x	x	
Desirable Security Requirements						
SR.2.1.1	x			x	x	
SR.2.1.2						
SR.2.1.3						
SR.2.1.4						
SR.2.1.5	x			x	x	
SR.2.1.6						
SR.2.1.7						
SR.2.2.1	x			x	x	
SR.2.2.2						
SR.2.2.3						
SR.2.2.4						
SR.2.2.5						
SR.2.2.6	x			x	x	

Table 10: Use Case #3 – Mapping Use Case Requirements to Technical Requirement

6.1.4 Horizontal Requirements for all Applications

For the definition of the MVP, in addition to the requirements applicable to the specific user stories that have been mapped above, there are a set of mandatory requirements (omitted from the tables) which are not tied to any specific user story. That is, there are no user stories who aim at demonstrating these generic requirements. Therefore, they should be regarded as transversal project requirements.

- *[TR.1.1.3]* It should support protocol and algorithm agility, i.e. the new TPM must allow application developers to choose among various cryptographic primitives and protocols that provide a specific service
    - Even though no user story is devoted demonstrating this functionality, the ability to allow vendors to implement new cryptographic algorithms as needed was already present in TPM 2.0. This essential feature facilitates that the algorithms can be changed without revisiting the specification, should they prove to be cryptographically weaker than expected.
- *[TR.1.3.6]* Allow support for some legacy primitives/protocols.
    - Although there are three user stories that are associated to this requirement (HWDU.NA.1, HWDU.NA.4 and HWDU.NO.1), it can also be regarded as a general technical requirement for the FutureTPM project.
- *[SR.1.2.3]* Reach QS-Level 1 (post-quantum crypto)
    - WP2 is devoted to the identification of suitable state of the art algorithms and primitives to reach QS1.

Also, the three requirements under section 5.2.1.4 respond to standard good security practices on information security:

- *[SR.1.4.1]* Allow the protection of sensitive information;
- *[SR.1.4.2]* It should be hard for an adversary to learn the secret information required for any action (e.g., authentication, encryption, etc.);
- *[SR.1.4.3]* Credentials should be stored on user device and must be protected from eavesdropping/leakage.

6.2 MVP Definition

Following the analysis conducted in the previous subsections, the FutureTPM MVP will be build based on the following requirements set shown in the following table, ordered by the coverage of requirement by the number of use cases, and at a second stage by the number of user stories.

The FutureTPM MVP thus consists of a total of 37 requirements, that are suggested as the initial core requirements of the platform to be delivered.

#	Requirement ID	Requirement Description	Evident in all Use Cases	Evident in 2 Use Cases	Evident in 1 Use Case	Covered by No. of User Stories
Mandatory Requirements						
1	TR.1.1.3	It should support protocol and algorithm agility,	X			
2	TR.1.3.6	Allow support for some legacy primitives/protocols	X			
3	SR.1.2.3	Reach QS-Level 1	X			
4	SR.1.4.1	Allow the protection of sensitive information	X			
5	SR.1.4.2	It should be hard for an adversary to learn the secret information required for any action	X			
6	SR.1.4.3	Credentials should be stored on user device and must be protected from eavesdropping/leakage	X			

#	Requirement ID	Requirement Description	Evident in all Use Cases	Evident in 2 Use Cases	Evident in 1 Use Case	Covered by No. of User Stories
7	TR.1.1.4	It should support enhanced authorization (EA)	X			17
8	TR.1.4.1	Provide support for at least one major OS (e.g. Linux)	X			17
9	SR.1.1.1	Pseudorandom number generator	X			17
10	SR.1.1.2	Key generation and storage functionalities	X			17
11	SR.1.1.3	Hash functions	X			17
12	SR.1.1.4	MAC	X			17
13	SR.1.1.5	Symmetric encryption	X			17
14	SR.1.1.6	Digital signatures	X			17
15	SR.1.2.1	Support for possible QR-crypto candidates for each category (symmetric, asymmetric and DAA	X			17
16	SR.1.2.2	QR Support for signing, key exchange, attestation	X			17
17	TR.1.2.1	It should be feasible to implement the chosen post-quantum algorithms on platforms with restricted memory, while providing an acceptable performance	X			9
18	TR.1.1.2	It should provide a small set of platform configuration registers (PCR)		X		13
19	SR.1.3.1	Support software measurement (PCR extend) and measurement reporting (Quote), using QR algorithms		X		13
20	SR.1.3.2	Support remote attestation functionalities		X		13
21	SR.1.1.7	Public key encryption and key exchange		X		8
22	TR.1.1.1	It should provide non-volatile random-access memory (NVRAM) storage		X		5
23	TR.1.3.3	Development and testing of a software FutureTPM, including adequate support for the Trusted Software Stack (TSS)			X	10
24	SR.1.3.3	Support sealing and binding operations			X	5
25	TR.1.2.2	The selected crypto algorithms can be implemented securely on an identified platform,			X	4
26	TR.1.3.1	Selected algorithms should be chosen in such a way that it is possible to enhance the performance of the cryptographic calculations with a small hardware coprocessor, in particular:			X	4
27	TR.1.3.2	Selected post-quantum cryptographic primitives should be chosen to allow for a maximum reuse of hardware accelerator engines			X	4
28	TR.1.3.5	Development and testing of a hardware FutureTPM (evaluation board), including adequate support for the Trusted Software Stack (TSS)			X	4
29	TR.1.3.6	Allow support for some legacy primitives/protocols			X	3
30	TR.1.3.4	Development and testing of a virtual FutureTPM, including adequate			X	3

#	Requirement ID	Requirement Description	Evident in all Use Cases	Evident in 2 Use Cases	Evident in 1 Use Case	Covered by No. of User Stories
		support for the Trusted Software Stack (TSS)				
31	SR.1.2.4	Provide a crypto library with TPM backed keys implementing TLS with QR algorithms			X	3
32	SR.1.1.8	Direct Anonymous Attestation (DAA) [for SW TPM]			X	2
	Desirable Requirements					
33	TR.2.1.1	The efficiency of FutureTPM primitives and protocols should be similar or better than the ones currently provided by TPM 2.0.	X			17
34	TR.2.2.4	Developing a Python based API or Library that works with Django REST Framework and integrates the TPM developed		X		14
35	TR.2.3.3	Easy to port to existing architectures		X		13
36	TR.2.2.3	Trying to integrate existing APIs with any type of TPM (QR or otherwise)		X		7
37	TR.2.3.2	Easy to support on mobile and IoT devices		X		4

Table 11: FutureTPM MVP

Chapter 7 Summary and Conclusions

The deliverable at hand which is the initial deliverable of the project has worked towards the definition of the mandatory and desirable requirements steming out of the vision of the consortium for a QR TPM implementation and has also defined in more detail the three use cases of the project, going down into the level of User Stories, that will be used to validate the overall FutureTPM implementation that will be released by the project.

As a consequent step, the use cases have been linked with the technical requirements, and the MVP of the FutureTPM project has been defined, which will be used as a reference list of the requirements that should be at least covered by the platform to be developed, and that will be tested by at least one of the demonstrators.

This list, alongside with the detailed descriptions of the Use Cases and their User Stories, will be used as input to the project's architecture definition activities, as well as the design of the demonstrator's plans, and will act as reference material for any consequent design and development activity of the project.

Chapter 8 Bibliography

[1] European Cyber Security Organization (ECS). European Cybersecurity Strategic Research and Innovation Agenda (SRIA) for a contractual Public-Private Partnership (cPPP). June 2016. [Available Online]: https://ecs-org.eu/documents/ecs-cppp-sria.pdf

[2] ISO/IEC 11889:2009 (all parts) Information technology – Trusted platform module.

[3] ETSI. Quantum-Safe Cryptography; Quantum-Safe Threat Assessment (2017-03). http://www.etsi.org/images/files/ETSIWhitePapers/QuantumSafeWhitepaper.pdf

[4] Will Arthur and David Challener. *A Practical Guide to TPM 2.0: Using the Trusted Platform Module in the New Age of Security*. Apress, Berkely, CA, USA, 1st edition, 2015.

[5] TCG. TCG algorithm registry. Committee Draft, January 7, 2016.

[6] G. Proudler, L. Chen and C. Dalton. Trusted Computing Platforms – TPM2 in Context. Springer, 2014.

[7] Ariel Segall. Trusted Platform Modules: Why, when and how to use them. Institution of Engineering and Technology, 2016.

[8] Ernie Brickell, Liqun Chen, and Jiangtao Li. A new direct anonymous attestation scheme from bilinear maps. In Trusted Computing - Challenges and Applications, volume 4968 of LNCS, pages 166–178. Springer Verlag, 2008.

[9] L.Chen and J. Li, “Flexible and scalable digital signatures in TPM2.0,” in ACM CCS. ACM, 2013, pp. 37–48.

[10] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, STOC ’96, pages 99–108, New York, NY, USA, 1996. ACM.

[11] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. J. ACM, 60(6):43:1–43:35, November 2013.

[12] Ahmad Boorghany, Siavash Bayat Sarmadi, and Rasool Jalili. On constrained implementation of lattice-based cryptographic primitives and schemes on smart cards. ACM Trans. Embed. Comput. Syst., 14(3):42:1–42:25, April 2015.

[13] C. Dods, N. P. Smart, and M. Stam. Hash Based Digital Signature Schemes, pages 96–115. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[14] Agile Business Consortium. Requirements and User Stories. Available at:

[15] Wake, B. (2003). INVEST in Good Stories, and SMART Tasks Available at:

[16] SCRUM

[17] The KANBAN process-management and improvement method.

[18] Extreme Programming.

[19] Roetteler, M. and Steinwandt, R. (2015). A note on quantum related-key attacks. *Information* Processing Letters, 115(1):40–44. https://www.sciencedirect.com/science/article/pii/S0020019014001719

[20] Trusted Computing Group. (2016). TPM 2.0 Library Specification.

Appendix A

FutureTPM Partners' Vision

TEC Vision

TEC. Since TEC will be mainly involved, from the technical point of view, in the identification, design and development of TPM's security, respectively of quantum-resistant symmetric and asymmetric cryptographic algorithms/primitives, TEC's focus is on the quantum-resistance as well as the integration of the primitives in the TPM. Therefore, TEC's vision of FutureTPM is directed towards the development and deployment of a novel robust and physical secure anchor/RoT, which withstands attacks performed by quantum computers. This resistance should be gained by means of the advancement of common and well-used cryptographic primitives of today. However, the evolution of crypto primitives, respectively FutureTPM should not lead into quantum-resistance for years or a decade, but rather remains secure for a long term. Since the qubits increased over the past years (1998: 2 qubits processable; 2017: more than 50 qubits processable), the evolution of the quantum computing power (qubits increase over past years) should/has to be considered as well.

Besides the more or less technical vision of FutureTPM, TEC's economical vision includes the following points as well:

- Hardware security anchor according to new defined ISO standard
- Approved and Implemented by all major Platform providers
- Produced by world top four suppliers
- Certified HSM supported by open-source software stacks

SURREY Vision

SURREY. SURREY's vision of FutureTPM is to provide a new generation of TPM-based solutions, incorporating robust and physically secure Quantum-Resistant (QR) cryptographic primitives (formally verified), to ensure long-term security, privacy and operational assurance in the complex domain of future ICT systems and services (thus, it has to be compatible with prominent computing platforms). The goal is to enable a smooth transition from current TPM environments, based on traditional cryptography, to systems providing enhanced security through QR cryptographic functions, including secure authentication, encryption and signing functions, thus, turning the host device into a "hardened" security token that may also remain secure long- term against an enhanced threat landscape in quantum computing deployments. By designing an innovative portfolio of high-security QR algorithms for primitives like Key Management, Encryption, Signatures, Hash- Functions, Message Authentication Codes (MACs) and Direct Anonymous Attestation (DAA), FutureTPM will fill the perceived gaps in the current status of cybersecurity.

FutureTPM should also support reactive, run-time risk assessment on all phases of the system development lifecyle, considering the complete TCG software stack. This will enable the system to generate a secure root of trust that can be used for e.g., interacting with cloud services, accessing corporate services and performing banking and eCommerce transactions.

UBITECH Vision
UBITECH. UBITECH's vision of future TPM is a robust and quantum-secure TPM compatible with TPM2.0. Future TPM should also support risk assessment on all the phases of a system development lifecycle. The recent developments in quantum computing make security one of the main concerns of TPM2.0. Therefore, algorithms of future TPM must be update with Quantum-Resistant (QR) to ensure security, privacy and trust as a long-term and reliable hardware root of trust. Future TPM should also support both classical and QR cryptographic primitives and algorithms.

RHUL Vision
RHUL. The RHUL vision for the FutureTPM project involves the development of provably secure quantum-resistant algorithms to be included in next-generation TPMs. FutureTPM will also seek to improve the existing TPM 2.0 capabilities, including remote attestation, the root-of-trust, and resistance to different classes of attacks. In its work, FutureTPM will take into account a range of different types of adversaries, including remote attackers, physical attackers and advanced persistent threats. We envisage that the results of the project will lead to technology affordable by everyone, and that can be included in a very wide range of devices, such as personal computers, laptops, smartphones, Internet of Things devices and smart-cards. As with TPM 2.0, we envision many possible implementation scenarios for future TPMs, e.g. including discrete, integrated, firmware, software and virtual. A key strategic objective of FutureTPM is to contribute to standardization efforts at EU level within the TCG, ISO/IEC and ETSI. Because the TPM shares many functions in common with other widely-used devices, such as HSMs and TEEs, we envisage that the FutureTPM solutions can also benefit them. In terms of areas of application, we expect that FutureTPM will build upon and strengthen the security of traditional applications of the TPM, for example self-encrypting drives, platform integrity measurement and strong authentication, as well as improving the security of online services, such as online banking and Cloud storage. Our aim is that FutureTPM will design and develop devices and algorithms that can be widely trusted, both by individuals, organizations and governments; to help ensure this ambition is realised, we will embrace an open design philosophy. We also intend that the achievements of FutureTPM will help to address current criticisms of the TPM, notably that it can be used to help "track" user activities, e.g. by manufacturers, by providing strong and provable privacy guarantees. In particular, we advocate strong visibility of who owns, and has access to, the private keys (e.g., the private endorsement key) underlying the security of the future TPM. In addition, we envision that FutureTPM will provide the basis to enhance the security functionality of systems into which they are embedded, without limiting what software a user can run on these systems. Our vision is that this project will positively affect the life of everyday people by enabling them to achieve greater levels of trust in the security and privacy properties of their always-connected activities.

IBM Vision
IBM. As one of the founding members of the Trusted Computing Group and as a company that is heavily invested into building a full-scale quantum computer, IBM recognizes the need for a TPM that will offer long-term security and hence is based on cryptographic algorithms that are resistant against attackers that have a quantum computer at their hands. IBM uses TPMs for many purposes, virtual TPMs on servers, supported with HSMs, and physical TPMs on devices on "edge" devices to ensure authenticity of collected data.

IBM has a strong research history in cryptographic protocols related to TPMs, in particular was one of the main designers the direct anonymous attestation protocol (DAA). To address the cryptographic challenges posed by quantum computers, IBM Research has recently established a world-renowned research team dedicated to quantum-resilient cryptography. Drawing on this, and in cooperation with the FutureTPM partners, IBM wants to build a new TPM specification that will provide security against future threat scenarios and that can be used on the systems of today.

UB Vision

UB. Inherent to the name, the FutureTPM should be forward-looking and future-proof, in terms of both security and functionality. The TPM has a successful lineage but its applications and implementations have branched significantly into specifications for hardware, software and virtualisation. The FutureTPM should maintain a common framework for these varied platforms.

The core concern for increasing the longevity of the FutureTPM is quantum resistance. But adding quantum-resistance to current TPM functionality should not necessarily be the only goal. It is an opportunity to improve and expanding upon existing hardware anchored security options. In an increasingly competitive market, given the proliferation of HSA and TEEs, a broad scope of applications for the TPM will make it a competitive choice which will in turn support greater uptake (by manufacturers, implementers, service-providers and end-users). It would be desirable for the FutureTPM to bring together the fundamentals of what a TPM has historically been, what it could be (based largely on the use cases but also with the potential for other functionalities) and how it will continue to be secure (through, for example, quantum resistance, but also more in-depth security verification).

IFAT/IFAG Vision

IFAT/IFAG. The vision of Infineon Technologies is a TPM 2.0 compatible future TPM that supports the features that are used nowadays but being secure also in the post quantum era. To achieve a user acceptance of the future TPM, the new cryptographic calculations should not decrease the performance significantly. Therefore, new hardware coprocessors provide means for a fast and secure calculation. At the same time the provided solutions should not significantly increase the costs of producing hardware TPMs. Thus, efficient post-quantum algorithms and implementations are needed.

UL Vision

UL. UL wishes to design a new TPM that fulfils the following properties.

- Functionality: The new TPM must provide at least all the functionalities offered by TPM 2.0.
- Security: The new TPM should implement cryptographic primitives and protocols that are secure against adversaries with quantum computing capabilities. Additionally, security of cryptographic primitives and protocols must be guaranteed when they are composed and executed in arbitrary environments.
- Efficiency: The cryptographic primitives and protocols implemented in the new TPM must be practical.
- Compatibility: It is desirable that the new TPM be compatible with TPM 2.0. If for certain services that is not possible, or if this endangers security, a justification must be provided.

- Algorithm agility: The new TPM should provide various cryptographic primitives and protocols for a given service and should allow new primitives and protocols to be implemented and added in future developments.

S5 Vision

S5. Suite5 envisages the integration of FutureTPM technologies in the delivery of the S5Tracker solution which can guarantee at the highest levels privacy preservation and data security and confidentiality. The employment of the FutureTPM methods will ensure, in QR level that data owners remain anonymized and virtually unknown to the main infrastructure platform (S5Tracker), while providing their personal data, while at the same time it is possible to trust both the data fetching as well as the data receiving edges, eliminating identify spoofing and data forging incidents. Furthermore, FutureTPM at the side of data analyst may enable secure data experimentation and management, not allowing data to easily leave the platform. The long-term vision of S5 with regards to its benefits from FutureTPM is to offer better than TPM2.0 functionalities to clients, without compromising at high levels in performance and application cost.

INESC-ID Vision

INESC-ID. INESC-ID envisions a future version of the TPM in which the functionalities provided by the TPM 2.0 specification are made available with security against adversaries having quantum computing capabilities. Interim solutions should also be made available, where both classical and post-quantum cryptographic primitives are supported, provided such solutions do not considerably increase the cost of producing TPMs. Users may select which implementations to use for each functionality, based on their performance and security requirements.

INDEV Vision

INDEV. The FutureTPM project is an opportunity to look forward and proactively protect sensitive data and information against quantum attacks. The big plan is to integrate the deliverable with popular libraries and web application frameworks (eg Django Rest Framework & .NET Core) making it available to thousands of developers and everyday users. Our app and the testing APIs that we will be providing, will develop a real testbed for this new TPM, possibly setting new standards for the authentication and signing of various important or confidential assets.

UPRC Vision

UPRC. The vision of UPRC for FutureTPM is to deliver new methods and procedures for risk assessment, threat modelling and hardware/software security evaluation for QR algorithms implementations in TPM 2.0. As there is a shift to provide security in the hardware level, FutureTPM solution is directly linked to this trend of delivering hardware-based security implementations using TPM modules. UPRC, with its vast experience as security evaluator, it will expand its expertise on TPM hardware security and will strive for excellence in the delivery of a new QR resistant platform. Innovative open source toolsets for risk assessment of platforms will be major development outcome of the project, while new scientific results will facilitate high quality standards and scientific publications in this emerging area of quantum computing and information security.

HWDU Vision
HWDU. The FutureTPM project represents an opportunity for the partners to provide a solution for threats (quantum computers) that will arise in the not so far future. By implementing a quantum resistant TPM, the consortium will have the chance to better understand the challenges of modifying the TPM to support the new algorithms and the software using the TPM. HWDU expects that a quantum resistant TPM will be more suitable for integration into the equipment of telco operators, whose lifetime is greater than that of consumer and enterprise products. HWDU also expects that the discussion will focus not only on algorithms, but also on how to take full advantage of the TPM capabilities for providing better security guarantees and how to enhance the TPM intrinsic architecture for enhanced resiliency.