FutureTPM

D6.6

Validation Results, Performance Evaluation and Adoption Guidelines

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 / D6.6/ 1.0
Work package contributing to the deliverable:	WP 6
Due date:	December 2020 – M36
Actual submission date:	February 9th, 2021

Responsible organisation:	UBITECH
Editor:	Dimitris Papamartzivanos
Dissemination level:	PU
Revision:	1.0

Abstract:	Deliverable D6.6 provides the final report of FutureTPM consortium and aims to provide a concrete evaluation of the FutureTPM framework and its building blocks. This deliverable critically appraises the technical developments of the project, highlights the lessons learnt, with regards to the implementation, integration, operation and execution of the demonstrators, while it provides adoption guidelines when it comes to the integration of QR algorithms in a Future TPM.
Keywords:	Validation results, Adoption guidelines, Takeaway messages, project Impact Assessment

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

Editor

Dimitris Papamartzivanos (UBITECH)

Thanassis Giannetsos (DTU)

Contributors

Dimitris Papamartzivanos, Sofianna Menesidou (UBITECH)

Rogério Paludo (INESC-ID), Luís Fiolhais (INESC-ID), Leonel Sousa (INESC-ID)

Roberto Sassu, Silviu Vlasceanu (HWDU)

Fanis Sklinos, George Evangelogeorgos (INDEV)

Sotiris Koussouris (S5)

José Moreira (UB)

Georgios Fotiadis (UL)

Liqun Chen (Surrey)

Bertram Poettering (IBM)

Thomas Poeppelmann, Albrecht Michael (IFAG)

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.

Executive Summary

In the context of this deliverable, the FutureTPM consortium aims to provide a concrete evaluation of the FutureTPM framework and its building blocks, considering the technical achievements and results generated throughout the technical work packages of the project.

Towards future proofing the connected world of tomorrow, the FutureTPM project investigates on the challenges for establishing trust in decentralised system architectures, where there is an increasing demand on pushing the trust establishment from the backend of systems towards the edges of networks. In this direction, this deliverable elaborates on the value propositions of the FutureTPM project and summarises the technical artefacts, which were developed in the context of the technical work packages of the project.

Given the value propositions of the project, the 2nd Chapter of the deliverable provides an overview of the FutureTPM framework and its building blocks in order to provide a holistic overview to the reader before we proceed to the critical appraisal of the technical achievements and evaluation results of the project. In addition, Chapter 2 performs an impact assessment in the application domains of the use cases of the project. More specifically, the demonstrators elaborate on the benefits that the FutureTPM framework brought in the Secure Mobile Wallet and Payments, the Activity Tracking and the Device Management setups, but they also elaborate on the impact that the project's technical achievements may have in the corresponding application domains.

Chapter 3 focuses on the core technical achievements of the project, namely the development of the QR algorithms that empower the QR TPM. More specifically, Section 3.1 elaborates on our research, the specifications of the QR algorithms, and finally the consortium's recommendations on the cryptographic schemes that meet the security criteria for the PQ era. Based on the analysis on the QR algorithms, Section 3.2 provides adoption guidelines for the design of a QR TPM. More specifically, in this section we elaborate on the design of the three variants of the QR TPM, namely the software, the virtual and the hardware QR TPM, and we offer a thorough discussion on the implementation and integration challenges that the consortium faced in this endeavour. In addition, we summarise the measurements collected from the lab testing for the performance and memory requirements of the QR algorithms for each of the TPM variants.

Chapter 4 elaborates on the consortium's achievements towards the security modelling and formal verification of the TPM. More specifically, we provide details on the modelling approach we followed, and we elaborate on our decisions on the modelling approach and our revised plan on dividing the modelling to individual functionalities of the TPM instead of performing the formal verification to the TPM as a whole. Among other functionalities, we elaborate on the formal verification of the remote attestation scheme, which is a common denominator in all the use cases of the project. In addition, we offer an overview on the method used in the new property-based DAA model, which is a combination of a traditional game-based model and a Universal Composability model for the security modelling of such a complex protocol such as DAA. The section concludes by providing future directions for extending the security modelling to other TPM functionalities considering more application domains.

Overall, this deliverable critically appraises the technical developments of the project, highlights the lessons learnt, with regards to the implementation, integration, operation and execution of the demonstrators, while it provides adoption guidelines when it comes to the integration QR algorithms in a future TPM.

Contents

List of Figures

List of Tables

Chapter 1 Hardening the Cryptographic Stack: Intertrustability of Quantum-Safe Systems of Systems

As the world transitions towards the Quantum Computing era, this evolution besides the clear benefits it brings forth towards the creation and growth of truly innovative markets, it also poses a number of challenges (or rather makes old unsolved challenges urgent to be tackled with); with **security, privacy, resilience, and operational assurance** being some of the major concerns at both logical extremes of a network, namely the edge and the cloud. Such a transformation is currently ongoing, cementing Europe's vision on **secure quantum computing being the key enabler for supporting the realization of Next-Generation Smart Connectivity "Systems-of-Systems"** (SoS) towards the evolution of such safety-critical SoS from local, standalone systems into safe and secure solutions distributed over the continuum from cyber-physical end devices, to edge servers and cloud facilities.

Towards this direction, one key pillar is the establishment of decentralized roots of trust in system components, and using these roots of trust to establish and maintain trust relationships. Once a trusted community is materialised, secure community communications can be established and used to provide trusted community-wide system updates. Thus, using the concept of a trusted community, trusted communities of communities can be created within a SoS environment. Prominent examples include emerging decentralized ecosystems, such as vehicular networks, metropolitan or industrial IoT ecosystems, that will play both an important role in the formation of the financial landscape, in the next decade, as well as having a significant social footprint. *In this regard, it is of outmost importance to investigate for robust solutions which will safeguard this emerging landscape.*

This is considered as one of the main goals of FutureTPM towards **"security- and privacy-by-design"** solutions. In FutureTPM, by "security- and privacy-by-design", we understand all methods, techniques and tools aiming at enforcing security and privacy properties at both network and system (software) level from the conception while guaranteeing their validity in parallel. FutureTPM makes use of advanced property-based attestation and verification methods with the aim of allowing intelligent (unverified) system components and controllers to perform with a predetermined envelope of acceptable behaviour and a risk management approach extending it to a larger SoS. Since the required security and privacy properties depend on the system and application domain, understanding these requirements and being able to precisely define them is a prerequisite.

Cryptography, in this context, has been the main anchor for safeguarding information and systems security. However, as we move towards the quantum era, we need to investigate for **robust cryptographic primitives** that will be the fortress to hold against advanced adversaries with quantum processing capabilities. In fact, both academia and industry have already started to develop solutions and drive standardisation activities towards the design and development of new quantum resistant algorithms which can replace those that have been safeguarding the security of SoS for decades, but cannot cope with advanced quantum threat models.

Considering the above, it is of paramount importance to develop solutions capable of achieving the strict requirements for security, privacy and trust of the emerging ecosystems of both the near and distant future and will be aligned with the community's efforts to establish standards that will hold in the post quantum era. In this context, the **FutureTPM project has worked towards the design and development of a quantum resistant TPM that will be the main pillar towards the vision of the project for future proofing the connected world.** The first stepping stone in this line of research is the **selection and design of QR algorithms** that can be integrated in the most widely used trusted component, namely the TPM, and to **formally verify** the security properties of this new generation of trusted hardware components. This overall vision of establishing **trust aware service graph chains** is also supported by advanced security solutions that can complement a provably secure QR TPM and offer a holistic **Risk Assessment** framework that can safeguard decentralised architectures throughout their life cycle, starting from the design until the runtime phase.

In this line or research, and in light of the Quantum Computing landscape, the next section elaborates on the value proposition of the FutureTPM project. Based on these achieved innovations, in what follows, the deliverable also provides a critical appraisal of all the project's artefacts towards quantum resistant and trust extensions; leveraging root of trust capabilities of a newly designed QR TPM middleware that guarantees and simplifies the trust relationships between all layers in the SoS runtime stack, thus, providing strong security and trust claims on the trustworthiness of all service function chains of a decentralized ecosystem. The aim of this process is to enable the support of extended “trust aware service graph chains”, for highly complex environments, with verifiable evidence on their correctness and functional safety, from their **trusted launch and configuration** to the **runtime attestation of both behavioural and low-level concrete execution properties** about a system’s integrity and execution correctness.

1.1 Value Propositions of Future TPM and Summary of Research Activities

As became apparent, the 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. Therefore, building on current TPM environments (leveraging traditional cryptography), the goal to provide recommendations on the next-generation of technologies enabling advanced 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 fills the perceived gaps in the current status of cybersecurity.

As a first step in this direction, we have investigated technical and security, privacy and operational assurance requirements for the new generation of the TPM-based solutions that are secure against the future large-scale quantum computer attacks. We have also conceptualized three industry-driven use cases (Chapter 2), in the context of emerging application domains with various security and privacy considerations including the Fintech, Assistive Healthcare, and Device Management ecosystems that allow the validation of the project research results in real-world scenarios and how the overall FutureTPM solution can serve vertical industry needs.

We have also identified a set of QR cryptographic primitives which could replace all of the classes of crypto algorithms supported by the existing TPM technology (Chapter 3). Our selection is based on the state-of-the-art researches in the Post-Quantum Cryptography (PQC) field; for example, several digital signatures, asymmetric encryption and key exchange mechanisms are chosen from the latest round of the NIST PQC standardisation process. This set of selected cryptographic mechanisms were implemented in one of the three TPM environments; namely **SW-, HW- and Virtual-based TPM**. The suitability of their inclusion in a future TPM was further evaluated based on the implementation performance, security analysis and public reviews.

In this line of research, the FutureTPM consortium has also designed two lattice-based Direct Anonymous Attestation (DAA) schemes (Chapter 4). DAA is an important cryptographic primitive that was originally designed to support user privacy when using a TPM chip. The QR DAA research has not yet been covered by existing standardization efforts. We have implemented both of these two schemes in a software-based TPM environment. These were then evaluated considering the feasibility of their inclusion in a future TPM.

The security modelling and analysis of the existing TPM technology and the QR cryptographic mechanisms have led to the definition of a newly introduced verification methodology (Chapter 5), based on a “bottom-up” approach, in which the focus is on modelling the core TPM functionalities towards building chains of trust (instead of considering the TPM as a whole). This was also supported by the definition of a "*trusted platform command abstractions*" model. This represents a formal model of a TPM command that captures the actions of the trusted platform module, when the command is executed, in such a way that excludes the cryptographic operations carried out internally and replaces them with non-cryptographic approaches. We essentially developed a trusted abstract platform model consisting of a specific set of formally-specified primitives sufficient to implement the core TPM functionalities beyond the core crypto operations. Such an abstraction modelling can enable the reasoning about and comparing different TPM services under various adversarial models and for different security guarantees, excluding any possible implications from the leveraged cryptographic primitives. For trusted platform module implementers, such a representation can be considered as a golden model for the expected system behaviour. From the perspective of formally verifying trusted hardware components, this model can provide a means of reasoning about security and privacy (of offered services) without being bogged down by the intricacies of various crypto primitives considered in the different platforms.

The FutureTPM threat modelling, risk assessment, and runtime risk assessment are another important part of the FutureTPM framework (Chapter 2). Research has culminated to the development of holistic risk assessment engine capable of providing functionalities during both design-time, where an initial risk graph of all possible threats and risks are identified, and run-time, where the risk graph can be updated in order to achieve the required security, trust and privacy properties in the case of newly identified (e.g., zero-day) vulnerabilities.

We have implemented a wide set of selected QR cryptographic mechanisms, with detailed evaluation results already been made available. This has also been coupled with a rigorous testing in the context of the envisioned use cases. A final on-boarding of this set of enriched use cases, in the overall FutureTPM framework, led to very promising outcomes with regards to the validity of TPM-backed solutions to serve vertical industry needs.

1.1.1 Highlights and Core Achievements

In this context, achieved outcomes of the project include the development of artefacts that push the state of the art in the aforementioned core areas (targeted by FutureTPM) of **trusted computing, QR cryptography, formal verification and security modelling, remote attestation of properties, dynamic real-time risk assessment, and enforcement of self-learning adaptable policies.** With this, we claim that a SoS can withstand even a prolonged siege by a pre-determined attacker with quantum computing capabilities as the system can dynamically adapt to its security and safety state. This is substantially more flexible than traditional security mechanisms that often try to maintain and enforce pre-defined policies using rather static security mechanisms. Even more, FutureTPM's intelligent multi-layered framework allows a very high degree of automation, something that is definitely required in CPS and IoT scenarios where the mere number of devices will prohibit human intervention for security management.

To achieve this goal, FutureTPM models CPSoS and breaks them down to a composition of multiple heterogeneous devices. Each of the sub-components can have individual policies and security mechanisms. Such security policies and solutions need to be aligned with safety requirements. On the level of sub-components, we can detect anomalies, attacks, and tampering more easily than on the overall SoS level. By using assurance and attestation services, FutureTPM increases the level of trustworthiness, certifiability and integrity of the overall SoS. Not only does this include integrity of hardware and software, but it also includes integrity and correctness of data. **Overall, FutureTPM provides a very high level of operational assurance in integrity, security and finally safety of the CPS SoS, by leveraging QR crypto primitives, as it actively manages the system states by permanently engaging the involved devices in the security management cycle**. Core artefacts created by the FutureTPM consortium include the following:

- Initial analysis of the **quantum-security of the low-level cryptographic schemes**, which must be supported by the TPM, and **investigation of different classes of current state-of-the-art QR crypto schemes** (hash functions, block ciphers, digital signatures, public-key encryption, and privacy-preserving primitives). This analysis enabled the selection of the most prominent algorithms (as well as designing new privacy-preserving schemes) to be implemented in the various TPM environments, as specified in Chapter 3.

- **Selection** of the set of current **state-of-the-art and newly designed QR cryptographic primitives** (related to asymmetric crypto, symmetric crypto and primitives for privacy enhancement) to be implemented in the various TPM environments. Since the goal of the FutureTPM project is to enable the transition to QR-enabled TPMs, a wide range of algorithms was selected so as to better **identify challenges that need to be considered** and also provide an **enhanced evaluation of various QR crypto algorithms** when implemented in resource-constrained environments. The chosen candidates are reflected in Table 1.
- Definition of a verification methodology for jointly **modelling TPM trust and security that abstracts TPM functionalities;** *Trusted Platform Command Abstractions* model. In particular, models in this framework can serve as both a basis for reasoning about the **security of applications and systems that make use of the TPM**-with the TPM use cases standing as applications of particular interest; and for reasoning about the security of the TPM's mechanisms themselves. Further, the model is designed to be compatible with the **trust monitoring and enforcement mechanisms,** developed in FutureTPM, allowing one to verify the security of an application based on abstract trust models, to later instantiate them - in particular trust and adversary models - with specific run-time attestation mechanisms to obtain a more concrete security results. **This break-down of TPM ideal functionalities and services allows for a more effective verification process towards building a global picture of the entire TPM platform security modelling as a Root-Of-Trust**. These models are designed to be modular and amenable to extension by the community.
- The design, development and implementation of **novel attestation and verification methods as a means of assurance and trusted interoperability** between a wide range of SoS. *Not only does this include integrity of system hardware and software but it also includes correctness and integrity of mission critical and/or sensitive data*. Concrete milestones led to the design of two families of trust extensions and implementation of: (i) **Integrity Verification supporting two modes of operation in Attestation by Proof and Attestation by Quote** for verifying integrity correctness of deployed systems, and (ii) **Control-flow Property-based Attestation** scheme for attesting, during run-time, the execution behaviour, through extracting the Control-Flow and Data-Flow Graphs, of specific properties of interest and functional behaviours.
- **Formalized the notion of secure remote attestation towards trust aware service graph chains** and presented TAMARIN security proofs showing that our models satisfy the three key security properties that entail secure remote attestation and execution: **integrity, confidentiality, and secure measurement**. Furthermore, in order to model this service, we also considered additional TPM processes such as the creation of TPM keys, the Enhanced Authorization (EA) mechanism, the management of the Platform Configuration Registers (PCRs), and the creation and management of policy sessions.
- The design and implementation of **advanced monitoring and introspection functionalities for the dynamic tracing of a system's control- and information-flow graphs** (needed by the runtime attestation enablers) by leveraging the most prominent schemes of **extended Berkeley Filters (eBPF)** and **Intel PT** tracing capabilities. In FutureTPM, dynamic tracing functionalities were provided, as programmable components, enabling the continuous monitoring of kernel shared libraries, system calls, shared data and memory address space, etc., and the in-depth investigation of the systems' behaviour for detecting cheating attempts or if any type of exploits targeting the program and data memory. This provides the TPM with the compiled control- and information-flow graphs (CFGs & DFGs) that represent the runtime state of a remote device, against the configuration and execution properties of safety-critical components.
- The provision of a set of **continuous risk-assessment and management mechanisms** able to evaluate in **real-time** the existing risks of the entire Systems-of-Systems against advanced quantum threat models; including network operation and availability attacks, low-level system attacks and data privacy risks and propose the appropriate mitigation actions.

In this context, FutureTPM designed risk analysis methods that target all the phases of a systems development lifecycle, from design time to near real-time risk quantification of newly identified attacks. The FutureTPM Risk Quantification Engine is capable of identifying and tracking the relationships among the cyber assets of the SoS, considering the underlying chain of systems, compute and storage infrastructure and use them to efficiently calculate individual, cumulative and propagated risks, as well as recommend and apply mitigation actions for tackling identified cyber threats.

- Conceptualization of **three industry vertical FutureTPM use cases**, in the context of emerging application domains with various security and privacy considerations including the Fintech, Assistive Healthcare, and Device Management ecosystems, and the final on-boarding and evaluation of the holistic FutureTPM framework, leading to very promising outcomes with regards to the validity of TPM-backed solutions to serve vertical industry needs.
- **Side-Channel Analysis (SCA) and other forms of implementation attacks were investigated on how they impact the security of a TPM**; where the attacker is typically considered to have physical access to the attached physical trusted platform module chip. The security of trusted computing and enclaves (including services such as protection of the trusted secrets, link between the trusted platform and the underlying operating system, chain-of-trust establishment and key hierarchies & management) was thoroughly evaluated under side-channel analysis. First, the vulnerabilities of reference implementations of QR algorithms were thoroughly analysed in terms of both horizontal and vertical side-channel leakage. Then, we proceeded on demonstrating some new software-based glitching attacks that were identified throughout the duration of the project

Overall, the FutureTPM project has drawn a lot of attention from the trusted computing community towards increasing the trustworthiness of ICT services and products. Based on the information from the Trusted Computing Group (TCG), more than a billion devices already use the TPM technologies. Virtually all enterprise personal computers, many servers and embedded systems include the TPM. Furthermore, all related TCG technologies, such as self-encrypting drivers and network security specifications have been used by networking equipment and other devices. All of these systems and applications can directly benefit from the research results produced from this project.

Finally, except from trusted computing, FutureTPM also has a strong impact on other applications of applied cryptography in general. For any application, which requires **long-term security for data protection and user privacy**, it can follow the outputted research results and technical guidance from this project to make a smooth transition from today’s cryptographic mechanisms to post-quantum cryptographic solutions. All these solutions (as will be presented in the following sections) have been implemented and heavily tested, thus, enabling the improvement in performance and efficiency of cryptography beyond the state of the art.

Chapter 2 FutureTPM Framework towards Future Proofing the Connected World

Key Takeaways on the Next-Generation of QR Trusted Platform Modules

- The consortium designed three Quantum Resistant TPM variants, namely Software-, Virtual- and Hardware- QR TPM. Each variant integrates a different set of QR algorithms:
    - **Software QR TPM**: *Kyber, Dilithium, NTTRU, L-DAA, SHA3/SHAKE*
    - **Virtual QR TPM**: *BIKE, SPHINCS+, Rainbow*
    - **Hardware QR TPM**: *NewHope, BLISS*
- We elaborate on how the holistic runtime Risk Assessment framework, which embraces the newly introduced remote attestation schemes and advanced runtime tracing techniques, can be used to provide trust guarantees and operational assurance on QR TPM-enabled decentralized system architectures of the future.
- The envisioned use cases perform an impact assessment in various application domains and elaborate on the beneficial characteristics that the FutureTPM framework provides.

2.1 Framework Overview & Building Blocks.

As the concept of quantum computing is becoming a reality, there is an imperative need to sustain information and systems security to the topmost level. Within the next two decades the first quantum computer will be available, and as has been highlighted by both the academia and industry, the advent of quantum computing proses an emerging threat for several cryptographic primitives and schemes that nowadays serve as the main pillar in the provision of secure services. Hence, as aforementioned, the main objective of the FutureTPM project is to identify the challenges and the key points that need to be taken into consideration now, i.e., at the state where quantum computing is in its infancy and the community works towards the definition of quantum resistant cryptographic schemes, in order to benefit the Trusted Computing technologies with the design of QR roots of trust.

Nonetheless, another key motivation of FutureTPM is to highlight the **benefits of moving towards more decentralised architectures with the use of decentralised roots of trust, where there is a need to shift the trust assurance from the backend systems to the edges of the networks**. In fact, this is the current research trend, based on which several industry domains, among others the vehicular networks for connected cars, Industrial IoT for human-robot collaboration and smart Aerospace, redesign their technology offerings to meet the challenges of the future. This research movement is in line with the objectives of FutureTPM project and our endeavour of bringing the QR primitives in embedded systems, as FutureTPM highlights the need to thoroughly analyse and standardise such decentralised security architectures and protocols.

Other building blocks that complement a QR root of trust, towards the realisation of such decentralised and mission critical architectures are the Risk Assessment and a new set of security verification and remote attestation protocols. The aforementioned building blocks capture all the different phases during the entire operational life cycle of a computing system, i.e., from the initial deployment and the identification of critical assets and risks, to the actual run-time verification and protection against even zero-day threats and vulnerabilities.

Accompanying with the QR algorithm design and implementation, the FutureTPM project demonstrated all of the produced technical artefacts (i.e., QR TPM, Run-time Risk Assessment, and the Integrity Verification and Control-flow attestation algorithms) in the context of three use cases (namely ePayment, Activity Tracking and Device Management), which were used to validate the feasibility and performance of a QR TPM in these three selected real-world systems that may be affected by the advent of quantum computing as a threat to security. Thus, the following sections elaborate on these technical artefacts. After highlighting the project's building blocks, we also perform and impact assessment in the application domains of the use case demonstrators to highlight the benefits that FutureTPM brings.

2.1.1 QR TPM

The developments of WP5 were focused on the **implementation, evaluation and testing of three TPM variants, namely a Software, a Virtual and a Hardware QR TPM**. Figure 1 offers an overview of the implemented QR TPM environments and highlights the QR algorithms which have been integrated in each case. In fact, the consortium considered the implementation and deployment of different QR algorithms, considering the state-of-the-art in the literature, but also the ongoing standardization activities in the field. More specifically, our intention was to be aligned with the NIST's Post-Quantum Cryptography Program, which entered in the final round in July, 2020.

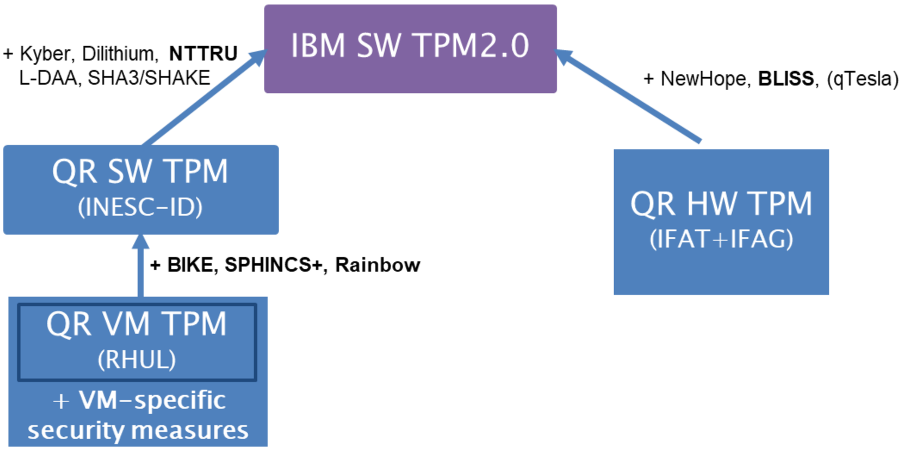


Figure 1: Overview of the QR TPM environments

Our motivation behind the implementation of these three QR TPM variants lies on the fact that our endeavour for future proofing the connected work should be aligned with the increased complexity and diversity of systems and software implementations. Hence, in order to ensure that FutureTPM will have a positive impact on a wide range of application domains, we proceed to the design and development of these three variants, so that to cover a wide range of requirements and possible applications embracing recent well-known technologies distributed over the continuum from **cyber-physical end devices (i.e., Mobile Edge Computing) to cloud facilities (i.e., Network Functions Virtualization).** Especially the latter requires the instantiation of such trusted computing technologies in virtualized environments for supporting privacy- and trust-aware service graph chains, in lightweight cloud-based ecosystems, with verifiable evidence of the integrity and correctness of the deployed containerized services.

Following this rationale, the consortium evaluated the applicability and performance of these three QR TPM variants in the context of the three demonstrators of the project, in order not only to investigate any integration and implementation challenges of the QR algorithms per se, but also to uncover the challenges posed in each application domain. In this direction, in Chapter 3 of this deliverable, we elaborate on our technical achievements and we provide adoption guidelines for the three QR TPM variants, while in Section 2.2 we perform an impact analysis for the application domains of each use case demonstrator.

It must be stated, that in order to meet the best practices in the process of the development of the QR TPM variants and the cryptographic schemes, such as remote attestation and Direct Anonymous Attestation which use the TPM as the trust anchor, we proceeded to the security modelling of the TPM and the formal verification of its security properties. These activities were carried out in the context of WP3. That is, Chapter 4 of this deliverable discusses a number of research challenges for the modelling of the TPM functionalities, and for capturing its usage in security protocols and the interaction of different parties with the TPM considering also the use cases of the project. **We believe that the produced models provide the baseline for an extensible verification methodology that enables rigorous reasoning about the security properties of Future TPMs.**

Based on the above, our research and development expands to several dimensions in order to cover adequately all the aspects needed for setting in motion standardisation activities and for pushing this new technology to the real world.

2.1.2 Run-time Risk Assessment

Towards our vision in future proofing the connected world by having a QR TPM as the trust anchor, the FutureTPM consortium investigated on complementary building blocks that can work in synergy with the QR TPM for enabling security and trust in the envisioned decentralised architectures.

In the face of an increasing attack landscape, it is imperative to ensure the correct and safe operation of all mission-critical business processes as, by their very nature, the internal physical and cyber (data and computing) assets—of an ecosystem or application domain—may not always be in trusted custody. Towards this direction, organizations must perform risk management so that they can identify and assess risks in order to keep them at acceptable levels. Thus, risk assessment serves as the foundation on which organizations can start building a well-rounded cybersecurity strategy.

The adoption of the risk assessment in the FutureTPM project aims to deliver a well-rounded framework that can be used to protect future decentralized system architectures using as a solid base the well-established methodologies of the field, but by enhancing them accordingly to meet the needs of the assessment of TPM-enabled architectures. More specifically, the use of TPMs enables a wide range of protective mechanisms of the trusted computing realm than can be used as possible countermeasures and extend the arsenals of defenders, in their effort to ensure that risk levels are kept under acceptable threshold. In this direction, our developments in the context of WP4 led to the definition a risk assessment conceptual flow shown in Figure 2 that can steer the assessment conduction during the whole life cycle of a system, starting from the design phase of the system and through its runtime phase.

The risk assessment process in the FutureTPM project aims to the definition and enforcement of attestation policies so that to check the correct operational state or the runtime behavior of the assessed TPM-enabled system. The whole framework is based on UBITECH's OLISTIC risk management suite, which is in position to maintain a digital reflection of the cyber-physical ecosystem in the form of interdependency graphs that hold together all the interrelated assets of the assessed environment. Based on this, the risk assessor is engaged in a semi-automated process where attestation policies are enforced to the deployment in order to attest its operational integrity using the TPM as the trust anchor in this process. Given the attestation outcome, the assessor can be sure for the operational correctness of the attested components of the deployment and in cases where the operational correctness is not verified, the assessor can proceed to the deployment of additional policies and checks to further investigate the security status during runtime. It must be stated that the development of the various risk assessment components, such as the newly introduced attestation by quote and attestation by proof schemes, the runtime tracing techniques, and the extension of UBITECH's OLISTIC risk management suite, was performed in the context of WP4 of the project. However, the integration of the risk assessment framework to the TPM-enabled system was materialized in the context of WP6, where through the components of the integrated framework, the risk quantification engine of OLISTIC works in synergy with the framework's policy decision and enforcement points to materialize the aforementioned life cycle. The interested reader can refer to D6.4 [21] for an overview of the integrated framework and for a detailed description of the workflow that connects the QR TPM and the risk assessment artifacts.

The convergence of the TPM that enables remote attestation with the risk assessment process through the dynamic operation of the TPM-enabled system is one of the key novelties of the FutureTPM project.

The next section elaborates on the remote attestation schemes and runtime tracing techniques designed in the context of the project to achieve Integrity Verification and Control-flow attestation.

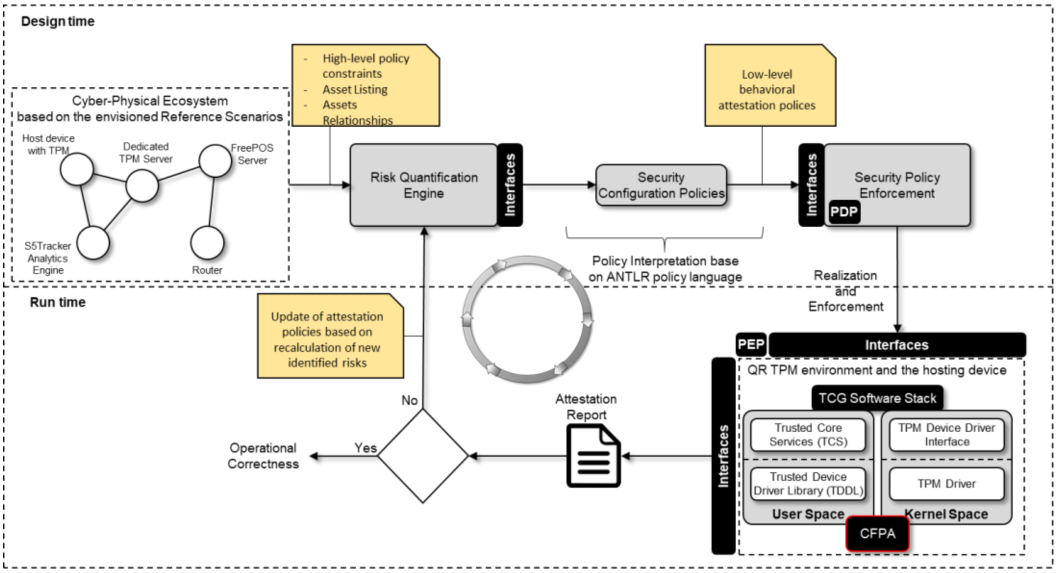

Design time
- High-level policy constraints
- Asset Listing
- Assets Relationships
Low-level behavioral attestation polices
Cyber-Physical Ecosystem based on the envisioned Reference Scenarios
Host device with TPM
Dedicated TPM Server
FreePOS Server
S5Tracker Analytics Engine
Router
Risk Quantification Engine
Interfaces
Security Configuration Policies
Interfaces
Security Policy Enforcement
PDP
Policy Interpretation base on ANTLR policy language
Realization and Enforcement
Run time
Update of attestation policies based on recalculation of new identified risks
PEP
Interfaces
QR TPM environment and the hosting device
TCG Software Stack
Trusted Core Services (TCS)
TPM Device Driver Interface
Trusted Device Driver Library (TDDL)
TPM Driver
User Space
Kernel Space
CFPA
Interfaces
Attestation Report
No
Yes
Operational Correctness

Figure 2: Conceptual flow of the runtime risk assessment

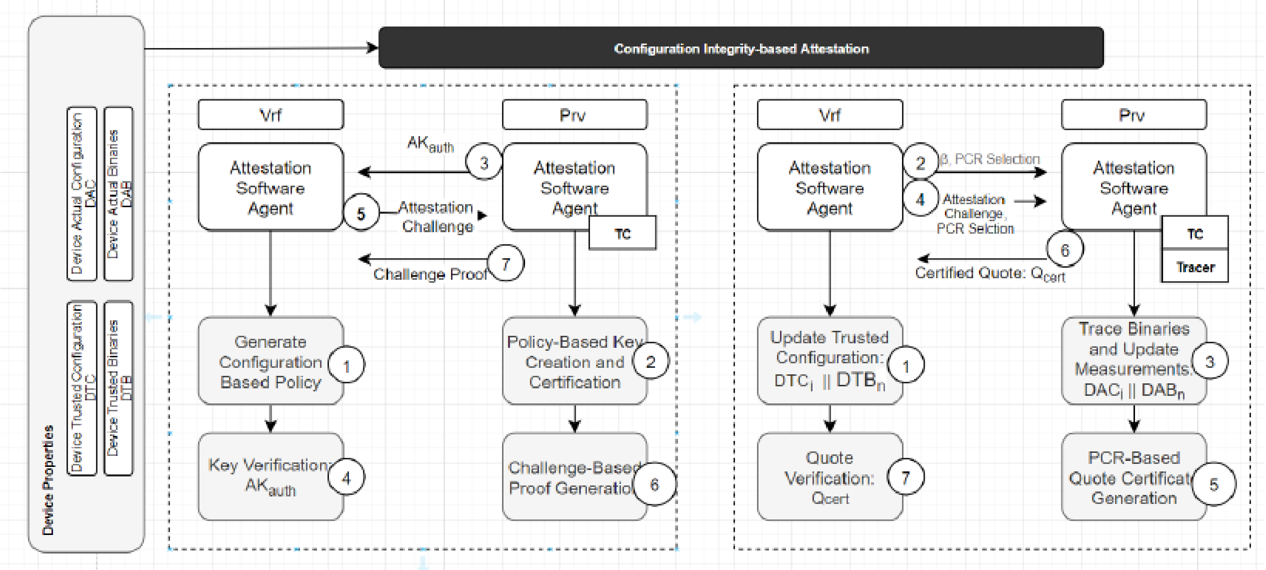
2.1.3 *Integrity Verification and Control-flow Attestation*

This section elaborates on our efforts towards designing **new remote attestation schemes and trust extensions as well as runtime tracing techniques to enable both Integrity Verification and Control-flow Attestation**.

Leveraging cryptographic techniques and Trusted Components (TC) towards protecting and proving the authenticity and integrity of computing platforms has been extensively researched. Both integrity and authenticity are two indispensable enablers of trust. Whereas integrity provides evidence about correctness, authenticity provides evidence of provenance. To achieve configuration integrity, there are two possible avenues: either make the configurations themselves immutable or make the hashes of the configurations immutable. The latter approach follows the Trusted Computing Group's (TCG) open integrity standards [3], which recommends the utilization of hardware TPMs for storing an accumulated hash over its Platform Configuration Registers (PCRs). TPMs can provide indisputable evidence of authenticity in the form of signatures over data using securely stored keys.

Integrity Measurement Architecture (IMA) [4] follows the TCG approach and accumulates measurements in a TPM. By default, IMA measures the load time integrity of user-space applications and files read by the root user during runtime. It is based on the Binary-Based Attestation (BBA) scheme proposed by TCG, where measurements and attestation consider hashes of binaries. However, even the smallest change in a binary dramatically changes its hash, making IMA measurements susceptible to grow unwieldy as the number of measured objects increases. Furthermore, the temporal order in which files are accessed, or applications are loaded, can be highly unpredictable, making it difficult to verify the accumulated measurements. The inherent disadvantage of BBA paradigms is the disclosure of the platform's software and hardware configuration, which is a legitimate privacy concern since an intermediate adversary (or a malicious verifier) can use this information to infer identifiable characteristics about the platform.

Further, the variety and mutability of software and their configurations make it difficult to evaluate the platform's integrity [5] during runtime. Several architectures extend upon the IMA-BBA paradigm to provide integrity verification like DIVE [6] and Container-IMA [7] . However, both solutions necessitate the exchange of some identifiable information. In the same line of research, Property-Based Attestation (PBA) [8] schemes map the platform configurations to attestable properties in order to avoid the disclosure of the host configurations altogether. The inherent limitation of PBA, however, is that it is only applicable to specific properties (which require accurate identification) and is not directly transferable to reflect changes of mutable configurations.



Configuration Integrity-based Attestation
Device Properties
Device Actual Configuration DAC
Device Actual Binaries DAB
Device Trusted Configuration DTC
Device Trusted Binaries DTB
Vrf
Prv
Attestation Software Agent
Attestation Software Agent
TC
AK$_{auth}$
3
5
Attestation Challenge
7
Challenge Proof
Generate Configuration Based Policy
1
Policy-Based Key Creation and Certification
2
Key Verification: AK$_{auth}$
4
Challenge-Based Proof Generation
6
Vrf
Prv
Attestation Software Agent
Attestation Software Agent
TC
Tracer
2
β, PCR Selection
4
Attestation Challenge, PCR Selction
6
Certified Quote: Q$_{cert}$
Update Trusted Configuration: DTC$_i$ || DTB$_n$
1
Trace Binaries and Update Measurements: DAC$_i$ || DAB$_n$
3
Quote Verification: Qcert
7
PCR-Based Quote Certificate Generation
5

Figure 3: Workflow of system Integrity Verification: Attestation by Proof (Left) and Attestation by Quote (Right).

In this context, the research activities of the FutureTPM consortium aimed to tackle the limitations identified in the state-of-the-art solutions and deliver a scalable and robust control-flow attestation schema. In this direction, D4.2 [13] redefined the attestation process by checking only the security-wise critical functions (instead of the whole application). This schema comprises three basic building blocks: the attestation protocol, the tracing technique, and the trust evidence collection mechanism. Based on this design, D4.4 [17] documented two newly introduced attestation schemes, namely the Attestation by Proof and Attestation by Quote (as depicted in the left and right side of Figure 3, respectively), while D4.5 [18] provided a thorough evaluation on the advanced multilevel tracing mechanisms that can enable the verification of the operational behavior of the attested system during runtime.

The key features of the attestation schemes are: i) the possibility for low-level tracing capability (Attestation by Quote), and ii) the capability which allows for attestation without disclosing any information that can infer identifiable characteristics about the individual configurations of the attested system (Attestation by Proof). The latter enables the integrity verification of a designated system without conveying additional or unnecessary information of the underlying host to a remote verifier, in case of a malicious verifier being aware of which components the underlying system have. This is of paramount importance especially in emerging environments with strict security, trust, and privacy requirements [9]. The offered integrity verification allows to assess and preserve the integrity of the deployed environment's Trusted Computing Base (TCB), at load-time and during system execution, by leveraging the capabilities of designed QR TPM implementations in the context of FutureTPM project, while ensuring predictability of the internal QR TPM PCR values regardless of the order of loading of applications/processes. It supports complete, configurable attestation that acquires binary signature chains from different unique registers, enabling advanced tracing capabilities to localize areas of compromise. The privacy-enhanced feature builds on the use of an Attestation Key (AK) within the TPM that can only execute a cryptographic operation if a set of PCRs is in particular (trusted) state, inferring the correctness of the component.

To materialise the Control-flow attestation documented in D4.3 [22], the consortium investigated on the extended Berkeley Packet Filters (eBPFs) and IntelPT tracing mechanisms for providing sound statements on the trustworthiness of the deployed devices and for collecting evidence to infer whether a system is under attack, during run-time. More specifically, eBPF is used for tracing the sequence of TPM commands and the binaries hashing for Integrity Verification. This is achieved by deploying hooks to the Linux kernel level, in order to intercept (hook) the TPM function invocations and resource accesses. More specifically, a C-based BPF interceptor traces the input and returned parameters generated by the invocation of the kernel functions. This process reveals the exact instructions and the corresponding parameters launched by the TPM Software Stack (TSS) to the TPM. Once these parameters are traced in the kernel space, they are parsed in order to match with the exact TPM commands to enable the analysis and to infer the use of possible weak primitives or the deviation of a normal behaviour due to the modification of the execution flow of an application. In addition, Intel-PT focuses to low-level mission-critical processes. Intel-PT is an extension of Intel Architecture that collects information about software execution such as control flow, execution modes and timings, and formats it into highly compressed binary packets. It is a relatively new kernel-based subsystem that provides, among other functionalities, a framework for hardware level analysis and is available to the 7th generation of Intel processors. In the context of FutureTPM tracing, Intel-PT is used in conjunction with the perf Linux analysis tool for better tracing capabilities. A thorough evaluation of the performance and scalability of the aforementioned tracing techniques can be found in D4.5 [18]. In addition, D4.5 we have identified the challenges and open issues that still need to be addressed in order to provide even more complete tracing solutions.

The **Integrity Verification and Control-flow Attestation novelties of FutureTPM**, were based on well-defined methodologies of the domain and provided several enhancements through the definition and implementation of **new remote attestation techniques** along with **advanced runtime tracing tools**. **These enhancements support the overall vision of the FutureTPM project to offer a holistic risk assessment methodology, tailored to the needs of future decentralised application domains in the PQ era.**

Given the above, the next section provides an impact assessment in the application domain on the use case demonstrators and describe how the demonstrators were benefited by the offerings of the FutureTPM framework.

2.2 FutureTPM Impact Assessment in Target Application Domains

This section aims to analyse the impact and the benefits that the FutureTPM framework brings forth to the application domains envisioned by the project's internal demonstrators. More specifically, the following sections elaborate on how the FutureTPM developments have contributed to form a new standpoint for each one of the demonstrators in the application domains of:

- **Financial Technologies**, via the Secure Mobile Wallet and Payments demonstrator.
- **Healthcare and physical activity data**, via the Activity Tracking demonstrator.
- **Network management**, via the Device Management demonstrator.

2.2.1 Demonstrator #1 – Secure Mobile Wallet and Payments

The Secure Mobile Wallet and Payments demonstrator aims to make the difference in the financial sector by enhancing the security levels of the offered mobile payment services for end-users. In fact, since mobile devices have become an integral part of our life, they facilitate a wide spectrum of financial services through the use of mobile Apps. Thus, the number of financial transactions executed in one-touch manner have led to a tremendous amount of financial data that pose strong confidentiality and integrity requirements, while the use of heterogeneous mobile devices, which can be exploited by adversaries in various ways, pose the need to invest on technology solutions that can provide high security and trust guarantees.

Enhancing security of mobile payment applications ultimately enhances consumer trust and this in turn will act as a catalyst for growth of mobile payments. This trust can be “earned” by providing visibility of the various security measures and controls, which are deployed to safeguard cardholders and customer's data privacy and identity. This visibility also helps mobile payment developers and mobile payment providers to engineer measures that can reduce the likelihood and impact of cyber threats exploiting vulnerabilities, weaknesses, and gaps in security controls of mobile payment applications.

Today mobile payment application development practices require to deliver mobile payment application software that is secure by design and by implementation. During design it is important to follow security by design principles. Specifically, for the design of secure mobile payment applications it is important to avoid design flaws that could impact the security of the mobile payment application and increase the risks of an attacker exploiting them to gain access to confidential cardholder data, confidential PII data and financial data. In fact, mobile devices have become complex environments that host software stacks and vendor-specific applications that bring vulnerabilities and flaws which extend the attack surface of the devices and threaten the trust bonds between the end-users and the service providers, due to the sensitive nature of the financial transactions.

More precisely, in order to enhance the security posture for the Secure Mobile Wallet and Payments domain, the demonstrator needs to offer the following qualities:

A. **Providing high security and trust guarantees at the device for sensitive financial data and transactions enablers** (authentication and financial tokens), so that the users can trust the Secure Mobile Wallet and Payments application and the financial services which are engaged in the transactions.
B. **Providing trust guarantees for the operational assurance of the mobile device and running applications**, so that to enable users validate the integrity of their device and applications before proceeding to financial transactions.
C. **Providing the necessary privacy and trust guarantees and protecting users' identity in a verifiable manner**, so that the users can be sure that financial service providers cannot track users' financial behaviour, but at the same time, users are hold accountable for their transactions.

Currently, even if there has been a notable growth of the financial technologies' domain empowered also by the increased capabilities of mobile devices' hardware and software, the security, trust and privacy guarantees highlighted above cannot be found in operational environments. This is mainly due to lack of standardised trust enablers in the vast majority of mobile devices, while due to the sensitivity of the Fintech domain, new technology offerings shall be approved by standardisation bodies and be widely tested and validated.

In light of the above, it becomes clear that the developments of the FutureTPM project can greatly benefit and further enhance the Secure Mobile Wallet and Payments domain, not only by integrating the exploitable artefacts of the project, but also through standardisation activities in the field that can drive the project research outcomes to production through innovation actions. The integrated framework of the FutureTPM project can be exploited to foster the developments in the domain, considering the generated knowledge towards addressing the challenges of integrating TPMs in mobile devices, the utilisation of Quantum Resistant cryptographic schemes in the context of TPM, and the development of remote attestation schemes and runtime tracing techniques in the context of a holistic risk assessment framework.

More specifically, the adoption of trusted computing practices in the financial technologies' domain is a major step towards meeting the aforementioned security and trust requirements and has a positive impact to the operational assurance of provided services. Based on the operational assurance, the adopters of trusted computing technologies can enhance their market position through the provision of financial services with strong security guarantees that can earn the trust of end-users who need to be sure for the security of their financial transactions. In this context, the Secure Mobile Wallet and Payments use case demonstrates the benefits of integrating a TPM component to serve as the main pillar for building additional functionalities, such as remote attestation and privacy preserving identity management schemes, which can have great impact in this business domain. In fact, the adoption the TPM in the context of the demonstrator highlights the need for encouraging standardisation actions that can lead to the adoption of standards in a horizontal manner for enhancing interoperability in the Fintech domain, but can set the basis for future vertical applications that demand the existence of strong trust guarantees.

Moreover, the integration of TPMs in the operation of Secure Mobile Wallet and Payments applications implies the **adoption of verified and provable secure cryptographic schemes**. In this way, mobile developers can capitalise on well-established schemes and avoid implementations that can bring flaws and lead to weak cryptographic schemes at the application level. Without doubt, such innovations can have a positive impact in the field, as it will lead to the development of mobile apps that can guarantee the secure and trusted communications, processing, and storage of data. On top of the clear benefit of the adoption of TPMs in the process, the research actions of the FutureTPM project, which focuses on the design and adoption of Quantum Resistant algorithms in the TPM, maximises the impact. In fact, the Secure Mobile Wallet and Payments demonstrator is not solely benefited by the use of trusted computing technologies, but notably has acquired the knowledge and experience of building security functions using quantum resistant algorithms that came to pave the way for the post-quantum era and offer applications that can stand against quantum threat models. Hence, the FutureTPM developments enable the demonstrator to foster the short-term developments of the field and establish its position in the market, but also has a great impact to the post-quantum roadmap of the fintech domain.

In our effort to assess the impact in the application domain, we must highlight the integration of the remote attestation schemes developed in the context of the risk assessment framework based on the use of the QR TPM. More specifically, the operational assurance of the Secure Mobile Wallet and Payments demonstrator is achieved using the Attestation by Quote and Attestation by Proof schemes for enabling the automatic, or upon request, secure establishment of trust between the Mobile App and the backend banking system. More specifically, the Attestation by Quote enables the integrity verification of the mobile device without conveying additional or unnecessary information of the underlying host to the remote verifier. The Attestation by Proof schema allows for attestation without disclosing any information that can infer identifiable characteristics about the individual configurations of the attested system. The attestation schemes are enhanced by the runtime tracing mechanisms developed and can take measurements of mission-critical functions on the attested systems. The combination of the aforementioned artifacts has a positive impact to the application domain, as remote attestation can be used to provide trust guarantees for the operational assurance of the mobile device and application configuration and runtime behaviour. Hence, the demonstrator can capitalise on state-of-the art solutions to ensure security and trust in future developments and service offerings will enable the beneficiaries to earn users trust.

Moreover, another important feature of the Secure Mobile Wallet and Payments use case was the utilisation of the **TPM and Direct Anonymous Attestation (DAA) to enhance the FIDO U2F Protocol for strong authentication**. More specifically, in the context of the demonstrator the consortium worked on the modelling aspects of integrating the TPM in the FIDO Protocol and addressed the challenges on how the DAA protocol can be used for achieving **unlikability of the end-user in the financial services domain**. By enabling both FIDO authentication and DAA in a unified manner, the demonstrator achieved authenticated and anonymous verification of Yubico credentials in the identity management process of financial transactions. This endeavour, which poses a significant research challenge in the field, led to the design of the overall system model and the documentation of the required trust models. The enhancement of the FIDO protocol has a twofold impact. On the one hand, the use of DAA can have a positive impact towards the privacy preservation of users. Through DAA the user can be authenticated to multiple financial services with different, but verifiable identities, so that to avoid footprinting of users' activity and protecting their fundamental rights and freedoms. On the other hand, the FutureTPM consortium aims to push the updated FIDO models to the FIDO standardization bodies for consideration to the future releases of the technical specifications. In fact, the latest FIDO working group has already identified DAA as a privacy preserving scheme that can benefit the FIDO protocol, however they have not released the technical details to achieve it. Hence, the enhancement of the FIDO protocol and the adoption of DAA can set the basis for the adoption of privacy preserving methods in the financial domain and give the advantage in the market by earning the trust of end-users.

Summing up, one can say that the financial technologies domain can be benefited by the integrated framework of the FutureTPM, through the provision of high security and trust guarantees for sensitive financial data and transactions enablers, the verification of the operational assurance of the mobile device and running applications, and the preservation of end-users' privacy in a verifiable manner. The integration of the QR TPM, as the trust anchor to build strong and provably secure financial services, can have great positive business and social impact not only in short term, but on the long run towards the post-quantum era.

2.2.2 Demonstrator #2 – Activity Tracking Demonstrator

The Activity Tracking demonstrator represents a set of applications and services that can be used to connect different individuals with organisations on the basis of (personal) data sharing. The most dominant domain on this direction, which is also the core interest of the Activity Tracking demonstrator is that of healthcare and physical activity data, where such services and infrastructures can be exploited to allow a flawless and close to real-time exchange of information between individuals (patients) and healthcare providers. The latter are actually the recipients of such data in order to effectively and timely monitor the condition and physical activity of users (patients), generate personalised recommendation and train individual as well as population-wide (based on their patients' base) machine learning models for advice offering, treatment and/or motivation.

As it is obvious the core interest for this domain, which is the actual power plant to such a system to operate, is that of keeping personal data safe during this journey, which starts with the collection of data by the individuals (users), the transfer to healthcare organisations and the final analysis which happens by data analysts that work on behalf of these healthcare institutions. As a consequence, the value that such a system can deliver relies fully and exclusively on the integrity of this chain of operations and therefore it is imperative to guarantee that the payload remains intact, exactly under the sharing modalities it is supposed to be and is treated based on the agreements signed between the different parties, keeping it away from malicious users and adversaries.

More precisely, as practise indicates, the needs to be satisfied by any infrastructure operating in this domain and is similar to that of the Activity Tracking demonstrator are the following:

A) **Providing high security and trust guarantees at the device, data and software system level** to safeguard that the payload is authentic and remains untampered and that all the necessary measures are in place to counter fight incidents which may lead to data loss, leakage, poisoning, etc., that could result to situations ranging from personal data exposure incidents, to the generation of false machine learning outputs which could ultimately mislead medical personnel in their judgements.

B) **Providing the necessary privacy and trust guarantees to the users** posting their data to the overall infrastructure, so that they can trust the platform which is receiving their data, be as confident as possible that their data will be handled as agreed in the supporting SLAs and data management agreements and reassure them that the degree of privacy which is directly linked with the sharing modes they have optioned for when they agree to share their data (e.g., the degree of anonymization chosen) is respected as it should.

However, despite the technical progress evident in many discrete areas in the data management and cybersecurity domain, the guarantees mentioned above are still not provided by operational systems as an inclusive offering, and especially in the healthcare domain, where spending and investments on ICTs have long been only considering the domain of prognosis and medical equipment and have only recently laid eyes on the domain of data handling and cybersecurity.

In this line, it is obvious that the introduction of an integrated framework as that of FutureTPM is in the position to positively impact the domain, **by providing the necessary tools and methods to not only allow the proper operation of a data collection and analysis infrastructure** such as the S5 Activity Tracker, but also to **accelerate the introduction of novel technologies and methodologies that allow the healthcare domain to speedily catch up with the mainstream security and trust solutions available in the market**, and overcome this state by becoming a pioneer in the application of novel methods (such as QR infrastructure). These actions are crucial as it is the value of the data and the critically of the systems that are integrated in the overall value chain (that is of the highest order and in many cases life-critical) which impose the needs for such investments.

In more detail, and regarding the first need identified above, the integrated FutureTPM framework is in a position to have a positive impact to the healthcare domain when it comes to the introduction of infrastructure that collects and handles **sensitive personal data**.

First of all, a positive impact can be immediately observed in the design phase of the infrastructure. The devices used for the collection, the transfer, the storage, the management and the analysis of the data and this the overall system can be checked prior to deployment and vulnerabilities that are present can be identified in order to optimise the infrastructure architecture, eliminate these weaknesses and seal the system against attacks that are based on known issues in the software or hardware of those entities. As such, a network of secure assets can be built, knowing that both the end-users (patients), the healthcare provider as well as the machines used by data analysts (in case the latter work outside of the premises of the healthcare provider) are secure and not prone to attacks. The result will be that the risks for integrating and deploying this infrastructure within the boundaries of a sensitive environment (such as the one of a healthcare organisation) is minimised.

The activities of the overall risk assessment are also able to reinforce the deployment actions of the infrastructure, as the risk graph will result to attestation and security policies that can be in place whenever a new device (Personal Activity Tracker or Analytics machine) is going to join the network. This is a quite crucial benefit for the overall system and the domain in general, as the overall network of devices is not something static, but the general idea (and ambition) is to make it expandable and ever-growing, in order to be able to offer the same services to more patients and generate a richer database to be able to run deeper analyses and identify patterns which may be only exposable if big amounts of data are available.

Moreover, the offerings of the integrated FutureTPM framework are also having an impact in a positive manner during the productive operation of the overall system. The same routines and probes do have a constructive effect to the monitoring and protection of the security of the system, as zero-day vulnerabilities can be spotted rapidly, as well as other kind of attacks, and mitigation measures can be launched, applying run-time security policies that can keep the system safe and trigger the necessary alerts to the system administrators for investigating such incidents and taking the proper decisions.

Furthermore, the attestation features allow the system to be able to identify that the data collection devices are not tampered with, and therefore that the veracity of data is quite high, thus there is no case of receiving poisonous data from the collection sources, which is quite critical for the overall value of the analysis to be performed, and in the same sense no un-authorised data collection and pushing devices can infiltrate to the network, as eventually their payload will be dropped as it will not be verifiable by the hosting platform.

Regarding the second expressed need, from the experiments performed and communication with domain experts and most importantly from end-users of the Personal Activity tracker, benefits are also expected for the domain. First of all, the introduction of QR-TPMs offering remote anonymous attestation (in our case with the invocation of the LDAA protocol) are contributing greatly to the construction of trust relationships between the users (patients) and the healthcare providers at the level of data sharing. By means of the different attestation features offered, data-owners are more relaxed as they can be reassured that they are providing the data, in a secure manner, to a trusted platform. Moreover, the same applies to the handling of their data, as they can be certain that the machines used to manage and analyse their data (for example a machine owned by a data analyst), conforms to detailed requirements set by the healthcare provider and is always attested and its integrity is checked before being allowed to connect to the system to fetch and analyse any data. In this line, when it comes specifically to the Activity Tracking demonstrator, and to similar approaches in the domain, it is important to be in a position to include into the overall ecosystem of its operation trusted devices. These are used at the edge of the infrastructure (e.g., at the data generation and collection points (see above), as well as the data analysis points), which in turn will provide guarantees regarding security and trust. These are considered highly important for the data that is being exchanged over the designed 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 are 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.

Moreover, another important feature that is provided through the TPM features embedded in the overall framework is that of **ensuring anonymization and privacy**, when the LDAA method is selected with the mode not to link the payloads send from a user with the user itself (for example by keeping the same base name for all users). Following this approach, on one hand it is guaranteed that the data that sent to the infrastructure are coming from a user who is allowed to connect and provide data to the infrastructure, and on the other hand it becomes impossible to track and identify this user. This feature is especially important when a user who wants to preserve his anonymity and thus his privacy but yet provide data to the system to contribute to the enrichment of the data pools held by a healthcare provider. Relevant to this, in the specific case of the Activity Tracking demonstrator, as indicated in previous deliverables, is the Data Anonymization and Privacy preservation service that is used to either secure the data and the details of each user to not be accessible from other parties accessing the platform, and also the generation of aggregated “User Personas” which are fictional representative users, that can be globally accessible by analysts, in order to create reference cases. As such, trust is the system (by the user) is also increased as the application of such methods allows to remain private, if so desired, but at the same time share data.

It needs to be noted that as identified also during the quantitative evaluation of the Activity Tracking demonstrator, the application of the LDAA protocol has a negative impact on the transfer times of data between the Personal Activity Tracker devices (patients) and the healthcare providers, which is inherent from the nature of the protocol, the strong security guarantees offered and of course is attributed to the nature of the quantum resistant algorithms that come into the picture. This delay, as shown during the experimentation with the revised version of LDAA (LDAA-v2) is however of the magnitude of some seconds, or in the worst case where the highest available security settings are selected reach the level of a couple of tens of seconds. However, this delay does not impose a blocking factor for the utilisation of the protocol. The first reason for this is that it provides, even when not using the highest security settings, a very high levels of trust, privacy and security guarantees which are absolutely necessary for the operation of such infrastructures. The second reason is that these delays are not critical for the operation of the Activity Tracking infrastructure, due to the system not relying, nor being based on the analysis of streaming real-time data, and as a consequence the delay in receiving data is not of significant importance. Apart from this, such delays are also expected to be drastically get reduced and become unnoticeable in the following years, as computing power will allow for faster execution of LDAA and similar QR algorithms.

Overall, one can argue that in the domain of healthcare and with the focus on the collection and analysis of personal sensitive healthcare relevant data, the use of the FutureTPM integrated framework comes as a **novel approach that can have a substantial positive impact when it comes to secure such infrastructures, preserve the privacy of the data and build relationships of trusts.** The FutureTPM integrated framework puts forth **trusted communication and information sharing between all entities** that take part in such a value chain and is able to provide an extra layer of privacy and security, handling both ex- and post-deployment issues which could lead to unwanted security incidents. At the same time, the inclusion of QR-TPM features in the framework are able to extend those benefits for a longer period than using the current TPM2.0 methods (with are nevertheless adequate at the time of writing this deliverable), as stronger security guarantees are offered, and the ground is prepared for making healthcare data management system resistant to quantum computing threats.

2.2.3 Demonstrator #3 – Device Management Demonstrator

Nowadays network infrastructures are becoming more and more important for the growth of the economy and to support the society. An issue in those infrastructures can cause a loss of connectivity and, consequently, a loss of money if a transaction cannot be completed. But more importantly, a malfunctioning network could cause a delay in the communication, which can be critical when people life is at stake. Thus, as we are relying on those infrastructures, we need to constantly check them to make sure that they work as expected or, otherwise, timely take the necessary countermeasures when a problem is detected.

Especially for mission-critical infrastructures, the network can be designed in a way that is able to face problems when they occur. A threat model can be made to clearly identify the weak points and solutions can be implemented to make the network more robust. Network administrators can count on a very wide choice of tools to help them to achieve the desired goals. However, often security people affirm that defensive security is way harder than offensive security. People working on defensive security have to be careful on every small detail; a small mistake could make protection worth millions of euro useless. People working on offensive security instead could achieve their goals easier and faster; a single weak point in the defenses could be sufficient to bypass them.

When a security incident happens, although the damage is already done, it is very important for defensive people to have an undeletable trace of the attacker's actions. This help first to assess the damage and, second, to understand why the security protections weren't effective and to improve them so that the attack cannot occur the next time. It is often a race between defensive and offensive people: defensive people work on making security protections more advanced; attackers work on making attacks more sophisticated. Both become better over the time.

The FutureTPM project, with the technologies it is based on and the ones it introduces, greatly helps defensive people to make their infrastructures more secure. In particular, for the network management use case, it addresses some weak points pointed out in D1.1 [14]: the lack of strong device identification due to the weak protection (by software) of the key used to prove the identity; incomplete monitoring of network devices, without taking into account software integrity; the lack of confidentiality and integrity protection on configuration data.

HWDU, a research-oriented European branch of Huawei, took the opportunity offered by the FutureTPM project to carefully design from scratch an advanced network management solution that solves the issues mentioned above. With this approach, we were able to shape the architecture in a way that security components are optimally placed, and we were able to focus solely on achieving the security goals without worrying about integration issues that likely arise when security features have to be added to an existing product.

The integration of FutureTPM-related technologies took place in two phases. During the first phase, we first analysed how the trusted computing technology could help to defend against the threats identified in D1.1 [14] and we designed and implemented a solution, that we call Comprehensive Integrity Verification (CIV) [43], which makes large use of the TPM 2.0 and its functionality. Subsequently, we replaced entirely the TPM 2.0 with the QR-TPM developed by the project and we made the necessary modifications to make use of the new quantum-resistant algorithms. With this approach, we were able to immediately take advantage of TPM functionality and to validate our design in an early stage of the project.

We believe that the results we obtained for the device management use case are satisfactory and we are also confident that our demonstrator will help Huawei and European companies to develop even more secure solutions than the ones available today. For the many companies that are not familiar with trusted computing, the FutureTPM project proves that this technology is ready for adoption into real products without too much effort. European citizens will certainly benefit if more companies use trusted computing as their network traffic will be handled by more secure routers. Also European service providers will benefit from this technology, as their devices will be protected with stronger defences, and they will be notified earlier and more precisely about security issues.

In addition to that, we gained a lot of knowledge from the integration of the QR-TPM into the demonstrator. We understood how to use the API to select the new quantum-resistant algorithms and where in the software stack support for those algorithms needs to be added. We found that, in order to use a QR-TPM, it is not sufficient just to choose the new algorithms when the functions of the library communicating with the QR-TPM are invoked, but also system libraries and applications need to be extended to support them. An example of such system libraries is Openssl, which requires support for the new algorithms in order to validate certificates for the TPM public keys. Also, low-level system components, such as the TPM driver in the Linux kernel and QEMU need to be modified in order to support larger keys. The details and the performance evaluation we provided in D6.3 [16] and D6.5 [19] will be helpful not only for implementers of solutions using quantum-resistant algorithms, but also for standardization bodies such as TCG to make the right choices for future specifications of a QR-TPM.

As a producer of equipment for telecom operators, Huawei will particularly benefit from the knowledge acquired with the demonstrator to plan in advance products with support for quantum-resistant algorithms. Especially in the telecommunication sector, the longevity of a product is particular important, as the product lifetime is usually 5 to 10 years, much longer than that of consumer products. Support for quantum-resistant algorithms, at least at hardware level, would increase the chances that the products will not become obsolete when conventional algorithms are broken. Although this is not expected to happen in the next years, vendors should already start the preliminary work necessary to support the new algorithms, as it might require several years.

Overall, with the good synergy among the partners of the project, we were able to explore not only from the theoretical but also from the practical point of view this novel area of quantum computing. The very good knowledge of all people involved made it possible to solve all the difficulties and to offer a concrete and working solution that incorporates the results from the research community and at the same time can be taken as a reference for further development by the industry.

Chapter 3 Next-Generation QR Trusted Platform Module

Key Takeaways on the Next-Generation QR Trusted Platform Module

- The consortium offers recommendations on cryptographic schemes that meet the security criteria posed in the PQ era. With respect to the asymmetric primitives, all four schemes (Kyber, BIKE, Dilithium, and SPHINCS+) are finalists of the ongoing NIST competition and standardization effort. The consortium also proposes five asymmetric primitives as secondary back-up choices (NewHope, NTTRU, BLISS, Rainbow, and Picnic).
- The consortium identified a set of integration and implementation challenges when it comes to the implementation of QR algorithms in the QR TPM variants. The current TPM specifications needs to be extended in order to meet the memory requirements set by the QR algorithms.
- The chapter summarizes the memory requirements and performance metrics of the integrated QR algorithms. The overall evaluation advocates that the vast majority of the QR algorithms achieve satisfactory results.

3.1 Suggestions for QR algorithms

3.1.1 Introduction

The work package WP2 of the FutureTPM project is dedicated to identifying suitable cryptographic building blocks for composing trustworthy TPM implementations in the presence of quantum adversaries. In a series of three deliverables (D2.1—D2.3), WP2 makes specific recommendations for concrete symmetric and asymmetric cryptographic primitives that, by the current state of knowledge, withstand quantum attacks. In this section, for the sake of completeness of the current deliverable D6.6, we reproduce partial results of the final deliverable D2.3 [20] of WP2.

3.1.2 General Methodology

The approach of WP2 to derive the project's set of suggestions was to closely follow the academic progress in the domain of quantum cryptanalysis, to study what the impact of new attacks on classic TPM primitives is (e.g., on encryption schemes, authentication codes, i.a.), to study which new algorithm proposals with the explicit goal of delivering quantum resilience have been made, to enquire with the FutureTPM teams of WP5 and WP6 that are responsible for implementing TPM prototypes about possible technical constraints, and to distil from all these inputs a small set of promising primitives that simultaneously meet the properties enumerated above. When doing so, the authors of D2.3 explicitly considered also the following key dimensions of evaluation:

- **Security.** The recommended primitives should deliver at least the same level of security against quantum adversaries as the established TPM primitives deliver against classic adversaries. By aligning their research with an ongoing standardization effort by the NIST, the FutureTPM project could directly benefit from a vast amount of cryptanalytic results on their proposals.

- **Availability.** It is a priority to have open-source implementations of the D2.3 [20] recommended schemes available, for many different platforms (desktop CPUs, microcontrollers, custom hardware, etc.). Having such implementations at hand greatly reduces the risk of obtaining erroneous (and possibly insecure) implementations, and it may help avoiding possible legal IP-related restrictions.

- **Adoption.** Presenting industry-adoptable solutions is a main goal of the project. Most of the recommendations made by WP2 are aligned with recommendations by the TCG (where applicable) or the NIST, or similar recommendations expected to be announced by the NIST in the near future.

3.1.3 Results

In our final recommendation, as communicated in Deliverable D2.3 [20], WP2 proposes to rely on the cryptographic primitives listed in the table below. With respect to symmetric primitives like hash functions and block ciphers, the selection criteria were mainly to use widely established primitives where digest sizes, key sizes, and block sizes are at least 256 bits. (The block cipher AES-256 meets this requirement only partially, see D2.2 [27] and D2.3 [20] for a discussion on this.) With respect to modes of operation like HMAC and CFB, the main selection criteria were to keep them in line with prior TCG recommendations as much as possible. (Of course, the operation of HMAC and CFB is conditioned on instantiating them with hash functions and block ciphers according to the list of recommendations.) With respect to the asymmetric primitives, all four schemes (Kyber, BIKE, Dilithium, and SPHINCS+) are finalists of the ongoing NIST competition and standardization effort, and they were picked with the aim of diversifying the underlying assumptions (lattice based, code based, and hash function based). Finally, D2.3 also proposes five asymmetric primitives as secondary back-up choices (NewHope, NTTRU, BLISS, Rainbow, and Picnic). We refer the reader to D2.3 for the symmetric back-up choices and more details on the selection rationale.

Table 1: Recommended schemes that meet the security criteria.

Category	*Recommended schemes*
Mandatory schemes:	
Hash functions	SHA-256, SHA-384, SHA-512, SHA3-256, SHA3-384, SHA3-512
Block ciphers	AES-256
Symmetric authentication	HMAC
Symmetric encryption	CFB
Public key encryption	CRYSTALS-Kyber, BIKE
Signature schemes	CRYSTALS-Dilithium, SPHINCS+
Optional schemes:	
Symmetric schemes	See Deliverable D2.3 [20]
Public key encryption	NewHope, NTTRU
Signature schemes	BLISS, Rainbow, Picnic

3.1.4 Discussion

The schemes recommended in the table were carefully evaluated in WP2, in many cases in joint work with other work packages of this project, and we **confirmed their suitability and applicability for use in a Future TPM**. *We emphasize that this does not exclude other options*. Indeed, many recent proposals in the domain of symmetric cryptography specifically target quantum-resilience, and all finalists of the ongoing NIST competition for (asymmetric) post-quantum cryptography might be good candidates as well. We had to limit the number of schemes to implement for practical assessment, and we can thus recommend only the ones listed in the above table.

3.2 Adoption Guidelines for QR TPM

There are several branches of Post Quantum (PQ) cryptography. The FutureTPM project has studied and analysed four PQ cryptography variants within the context of the TPM: **lattice-based, hash-based, code-based, and multivariate-based.** The guidelines and recommendations provided in this section are derived solely from the SW-TPM and its accompanying stack, the TSS. Before providing any insights on the viability of each scheme, we need to answer the following outstanding questions:

- *Would PQ algorithms be realizable in a physical TPM? If so, what are the costs? (Memory, Domain Specific Accelerators)*
- *Is a PQ physical TPM cheap to manufacture?*
- *Are the new PQ algorithms able to replace the functionality provided by the current cryptographic algorithms? If so, at what runtime cost?*

3.2.1 Software-based QR TPM implementation

Figure 4 shows the basic architecture of the legacy SW-TPM as provided by current open-source implementations, where the Transmission Control Protocol (TCP) interface emulates the TPM Command Transmission Interface (TCTI) layer found in the physical TPM. The base architecture is composed by the following components:

- A cryptographic processor wherein a secure Random Number Generator (RNG), Rivest-Shamir-Adleman (RSA) and Elliptic-curve Cryptography (ECC) cryptographic primitives, and a hashing engine are available.
- A small persistent memory module (64kB) to store TPM's state.
- A versatile memory to keep short-lived data (generally implementation dependent).

Through the emulated TCTI layer, a client is able to interface with a SW-TPM using the commands provided in the TSS.

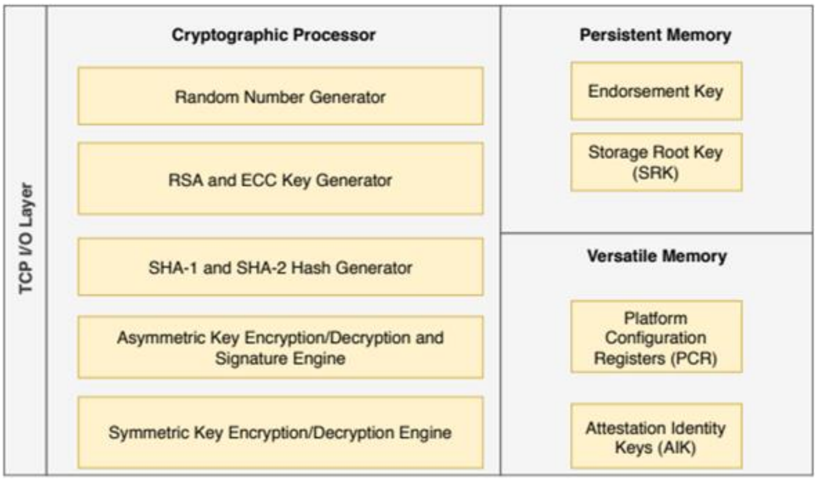

TCP I/O Layer
Cryptographic Processor
Random Number Generator
RSA and ECC Key Generator
SHA-1 and SHA-2 Hash Generator
Asymmetric Key Encryption/Decryption and Signature Engine
Symmetric Key Encryption/Decryption Engine
Persistent Memory
Endorsement Key
Storage Root Key (SRK)
Versatile Memory
Platform Configuration Registers (PCR)
Attestation Identity Keys (AIK)

Figure 4: Architecture of the SW-TPM.

3.2.1.1 SHA-3 Implementation

The single new addition to the hash generators module is SHA-3, as it is used by multiple PQ algorithms for Key Derivation Functions. SHA-3 support is provided in all expected endpoints with support for all variants. There was no need for new hardware structures or the addition of memory.

Besides the standard cryptographic functions, the SHA3 standard also provides Extendable Output Functions (XOF) labelled as SHAKE. SHAKEs are cryptographic hash functions able to output an arbitrary number of pseudo-random bytes. Since the main goal is to emulate a future hardware TPM, there needs to be a limit imposed on the size of the XOF output, commensurate with hardware implementations. Kyber and Dilithium make use on average of 547B from an XOF. The TPM uses a buffer equal to the maximum size of all supported hash algorithms, where the largest hash output by the standard commands is 64B from SHA512 (or SHA3-512). The standard TPM architecture supports a maximum buffer size of 2kB. This buffer size is used for sending messages to be encrypted through symmetric or asymmetric algorithms, sequence updates and performing Message Authentication Codes (MACs) to name a few. The chosen buffer size for XOFs must strike a balance between the current and future PQ algorithms, e.g., using the TPM as an accelerator for rejection sampling when building pseudo-random secret data, or when performing a TPM backed mutually authenticated Key Exchange (KEX). Therefore, in order to reuse the maximum buffer struct already available while attempting forward-compatibility for the use-cases previously referred, the upper bound of an XOF was set to 1kB.

3.2.1.2 QR Algorithms Implementation

The APIs modified and added by the algorithms implemented in the SW-TPM are described in Table 2. Every algorithm used the reference implementations provided by the consortium with the exception of LDAA which was implemented based on its reference implementation. Since the TPM does not provide vector units, the vectorized implementations were not used.

While the reference implementations only required integration in the TPM environment, the LDAA protocol required developing new software for two recent results from the on-going research on Lattice-based Direct Anonymous Attestation, namely [1] and [2]. The software for the LDAA protocol presented in [1] was developed first, but the results were impractical in terms of execution times and memory sizes due to the complexity of the underlining arithmetic operations and the overall scheme. To improve on these issues new research revealed a new improved LDAA version presented in [1], which is mainly aimed at reducing the sizes of the overall protocol whilst shifting most of the heavy signing and verifying computations to outside of the TPM environment. The software implementation of the new protocol was revealed to be problematic in the TPM due to the size of the prime modulus (around 70 bits as suggested in [2]). *While the RSA hardware accelerator could be used for such an implementation, its availability was not clear in a PQ hardware scenario at the time.* It could be the case that a PQ TPM would not have support in hardware for RSA. Therefore, a full software version was developed without using any external libraries and without relying on such an accelerator. The choice of the ring in the algorithm also increases the complexity of the polynomial multiplication, which is the basic arithmetic operation for the entire LDAA scheme, since it is an unfriendly modular ring it cannot be based on the Number Theoretical Transform. The results in Table 3Table 5 already consider the improved version of the LDAA [2].

Since the LDAA protocol also involved the development of the software for the Issuer, Host, and Verifier, outside of the TPM environment, most of the same issues faced on the TPM were present on the implementation of these instances. The difference, however, is that such software would run on general purpose machines where multiple-precision libraries are available. Nonetheless, due to the immaturity of the scheme and lack of concrete parameters, the signatures generated are currently always failing to be verified. On the new scheme, the verifier checks whether a signature on a given message concerning a basename is valid, which entails checking several equalities that should remain true under automorphism stability. The current implementation, however, is failing to verify the automorphism. The issue could be related to the specific description provided in [1] or an implementation issue. Since this issue is of current research interest, it is still under investigation and should be corrected as future work. Even though concrete execution times were not measured for the verify command on the new scheme, it is expected that it should be faster in comparison to the previous version of the LDAA. This follows the speedup observed in computing the signatures, according to the results presented in D6.5 [19].

Even though the version of the LDAA [2] improves on the initial research results presented in [1], it is still a far more complex scheme in comparison to the remaining PQ algorithms assessed in the software TPM. **This raises questions regarding the feasibility of a TPM-based LDAA**, especially comparing the results presented in Table 3 and Table 5. It is important to emphasize that this is still a novel research field and was a late addition to the tasks. As such, the results presented herein are not optimized but could potentially be improved in performance contributing to a more practical scheme whilst also being corrected for its full functionality.

Table 2: QR Algorithms and corresponding endpoints

	Hard Problem	*Modified Endpoints*	*New Endpoints*
Kyber	Lattice-based	TPM2_Create TPM2_Load TPM2_CreateLoaded TPM2_CreatePrimary TPM2_LoadExternal TPM2_StartAuthSession TPM2_Duplicate	TPM2_KYBER_Encrypt TPM2_KYBER_Decrypt TPM2_Enc, TPM2_Dec
Dilithium		TPM2_Create TPM2_Load TPM2_CreateLoaded TPM2_CreatePrimary TPM2_LoadExternal TPM2_Sign TPM2_VerifySignature TPM2_Quote TPM2_Certify TPM2_Duplicate	---
NTTRU		TPM2_Create TPM2_Load TPM2_CreateLoaded TPM2_CreatePrimary TPM2_LoadExternal TPM2_StartAuthSession TPM2_Duplicate	TPM2_NTTRU_Encrypt TPM2_NTTRU_Decrypt TPM2_Enc, TPM2_Dec
LDAA		---	TPM_CC_LDAA_Join TPM_CC_LDAA_SignRequest TPM_CC_LDAA_SignProceed

Regarding Kyber, Dilithium and NTTRU algorithms, they fit into the already available TCG 2.0 specification. They only require the addition of new flags for each algorithm and the creation of their respective Algorithm IDs. We also provide four new encryption and decryptions endpoints for Kyber and NTTRU in order to provide feature parity with RSA. Further, we also offer two new endpoints for encapsulation and decapsulation.

Table 3: QR Algorithms and memory usage

	Memory Usage in kB (Max)			New Hardware Structures
	Versatile	**IO**	**Persistent**	
Kyber	18 (KEM_Enc)	3.6 (Decrypt)	Increase proportional to size of private key	No new hardware
Dilithium	102.6 (Sign)	5.7 (Keys)	Increase proportional to size of private key	
NTTRU	6.6 (KEM_Dec)	3.3 (Decrypt)	Increase proportional to size of private key	
LDAA	1300 (Sign)	1600 (Signature)	1600 (private key)	

Table 4: QR Algorithms and key size details

kB	Public Key	Private Key	Signature/ Commit	Encryption	Encapsulation
RSA (2048 bits)	0.29	0.27	0.27	0.26	---
ECC (nitsp256)	0.14	0.10	0.08	---	---
EC-DAA	0.14	0.10	0.25	---	---
Kyber768	1.10	2.50	---	1.20	1.09
Dilithium (III)	1.50	3.60	2.70	---	---
NTTRU	1.25	2.50	---	1.30	1.25
LDAA	32.8	65.7	1300	---	---

Table 5: QR Algorithms and overview of timing performance measurements

ms	Key Creation	Signature/ Commit	Verify Signature	Encryption	Decryption	Encapsulation	Decapsulation
RSA (2048 bits)	275.78	166.84	165.18	165.54	166.39	---	---
ECC (nitsp256)	169.49	167.03	167.06	---	---	166.81	337.79

ms	Key Creation	Signature/ Commit	Verify Signature	Encryption	Decryption	Encapsulation	Decapsulation
EC-DAA	170.05	166.96	---	---	---	---	---
Kyber768	166.56	---	---	166.06	166.67	167.57	166.08
Dilithium (III)	166.52	170.53	166.79	---	---	---	---
NTTRU	166.05	---	---	165.53	166.30	167.05	166.11
LDAA	335.16	38336.43	---	---	---	---	---

Table 3 shows the maximum memory usage for each algorithm as well as new hardware structures required for its correct operation which the TPM did not previously provide. Table 4 compares the TPM outputs between PQ lattice-based algorithms with RSA and ECC. The executed time tests are presented in Table 5. The ASCII string *"My super secret. Please don't share.\n"* is used for encryption and signature; signed messages use the SHA3-256 hash; and all keys are created as non-primary with the fixed TPM and parent properties. All the measured times result from taking the median over one hundred runs running on an Intel Xeon Gold 6140 at 2.3 GHz.

Regarding the lattice-based algorithms, Dilithium requires the most memory with 102.6kB in versatile memory and 5.7kB in IO memory. We should note, however, that these results are for the upper bound of memory requirements, i.e., we are using the most secure parameters. A physical implementation of a PQ TPM will most likely only provide support for the smaller recommended parameters. Kyber and NTTRU are within the expected range of required memory.

When comparing the memory requirements of lattice-based PQ algorithms with state-of-the-art algorithms (RSA and ECC), the comparison is unfavourable. Table 4 shows an increase of one order of magnitude when directly comparing its outputs using the same inputs.

Comparing the key creation execution time, the QR algorithms show a speedup over RSA of 1.23x and a small slowdown of 0.95x in relation to ECC, specifically Kyber and Dilithium. Regarding signature and encryption/decryption execution times, the PQ schemes show commensurate execution times in comparison to their traditional counterparts. However, if the application constraints allow it, the addition of a vector unit would provide speedups of 3x to the lattice-based algorithms in the QR TPM.

3.2.1.3 QR TPM Architecture

The final QR TPM architecture is shown in Figure 5. **The new architecture adds a new module for SHA3 hashes, a lattice-based engine in order to support Kyber, Dilithium, and NTTRU, and additional memory to support LDAA.** Based on Table 3 the required memory footprint for to support LDAA would be around 1.6MB for persistent memory, 1.3MB for versatile memory, and 1.6MB for TCP/IO buffers, which is more than sufficient to support Kyber, Dilithium and NTTRU.

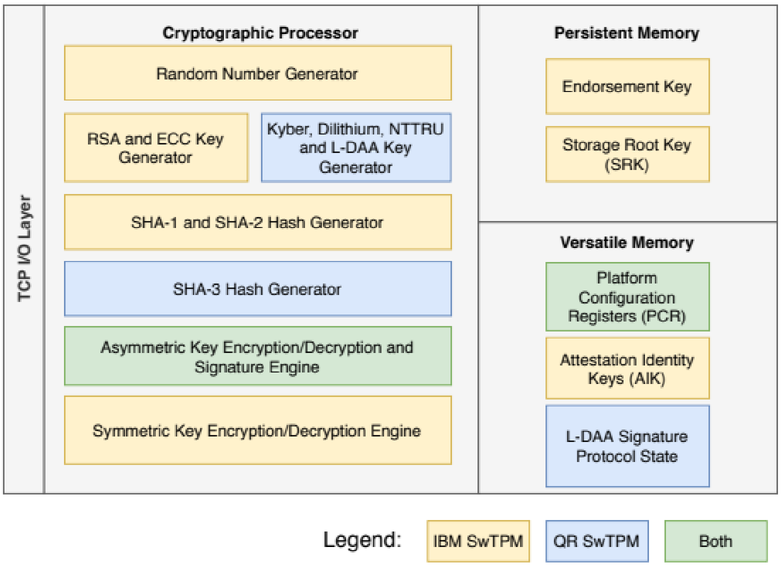

TCP I/O Layer
Cryptographic Processor
Random Number Generator
RSA and ECC Key Generator
Kyber, Dilithium, NTTRU and L-DAA Key Generator
SHA-1 and SHA-2 Hash Generator
SHA-3 Hash Generator
Asymmetric Key Encryption/Decryption and Signature Engine
Symmetric Key Encryption/Decryption Engine
Persistent Memory
Endorsement Key
Storage Root Key (SRK)
Versatile Memory
Platform Configuration Registers (PCR)
Attestation Identity Keys (AIK)
L-DAA Signature Protocol State
Legend:
IBM SwTPM
QR SwTPM
Both

Figure 5: QR TPM Architecture

3.2.1.4 Discussion & Critique

Experimental results suggest that the benefits of adding Kyber, NTTRU, and Dilithium to a hardware TPM far outweigh their cost from a security and resources perspective. In addition to their execution times having similar costs to traditional cryptography, the implementation of a single vector unit brings performance improvements to both schemes. The addition of a vector unit may come with certain underlying costs. Depending on the width of the vector unit we would need to use more on-chip area which, in turn, increases the temperature, the TDP, and the cost of the chip. There needs to be further research in this area to ascertain what the ideal width of the vector unit is in order to balance out the three variables. From a memory perspective, there is a significant increase, which in turn increases the area required by the TPM and its cost. However, we believe the added security offsets the memory cost.

The implementation of QR algorithms highlight how insufficiently equipped the current TPM architecture is when using lattice-based algorithms. The addition of a vector unit would net a 3x speedup to most lattice-based primitives. Alternatively, the addition of DSAs for polynomial arithmetic would greatly benefit the TPM.

However, the addition of QR schemes to the TPM should not be regarded as an all or nothing scenario. Rather, the proposed schemes should be carefully chosen and slowly rolled into the TPM architecture. Three great candidates are: Kyber, Dilithium, and NTTRU. Since they all share the same underlying arithmetic, the addition of one streamlines the addition of the remainder or future lattice-based proposals. Further, it can be concluded that Kyber and Dilithium can be deployed with marginal impact to the user. The TCG specification only needs to add new flags for certain endpoints and assign an algorithm ID to each algorithm.

3.2.1.5 Guidelines

When PQ algorithms are standardized by NIST, we recommend slowly rolling them out to users. **The hardware design phase should focus on a single algebraic backend such that, if future or current algorithms use the same backend, they can share the same memory buffers and hardware structures.** For example, Kyber and Dilithium, which share the algebraic backend, are both finalists in NISTs PQ competition and are excellent choices to start replacing the functionality provided by RSA and ECC. Once again, we envision minimal modifications to the TCG specification in order to support both Kyber and Dilithium. The addition of flags to certain endpoints and assignment of algorithm IDs should suffice.

The cost of adding PQ algorithms is only paid upfront from both a hardware and API perspectives. Once one lattice-based scheme is added to the TPM, the addition of new schemes will only require minimal modifications. The same occurs in the API. The TCG could ratify a specification which includes support for some PQ schemes which will be slowly rolled out to users. From a user perspective, the code remains the same but when calling a certain scheme, the user inspects the TPM's capabilities before proceeding.

3.2.2 *Virtual-based QR TPM Implementation*

Figure 6 presents the architecture of the Virtual-based TPM used in the context of the FutureTPM project. The consortium faced several technical challenges during the implementation phase of the QR V-TPM which are discussed in detail in Section 3.2.2.2 in order to provide adoption guidelines.

As described in deliverable D5.2 [25], and in the document "Technical Guide to V-TPM ", the V-TPM architecture uses a modified version of LIBTPMS which contains the post-quantum algorithms. It needs to be stated, that in this architecture, a Virtual Machine (VM) is connected using a driver to a TPM emulator (based on QR-Libtpms) by using TPM-TIS buffer. Each VM contains a QR TSS, which is the modified version of the TSS of IBM with the new functions which have QR-algorithms added in the virtual TPM code. The functions (*`createprimarykey`*, *`signed`*, *etc*) executed by the TSS, are captured by the Virtual-TPM driver. The Virtual-TPM driver is created by the hypervisor QEMU when the VM is powered on.

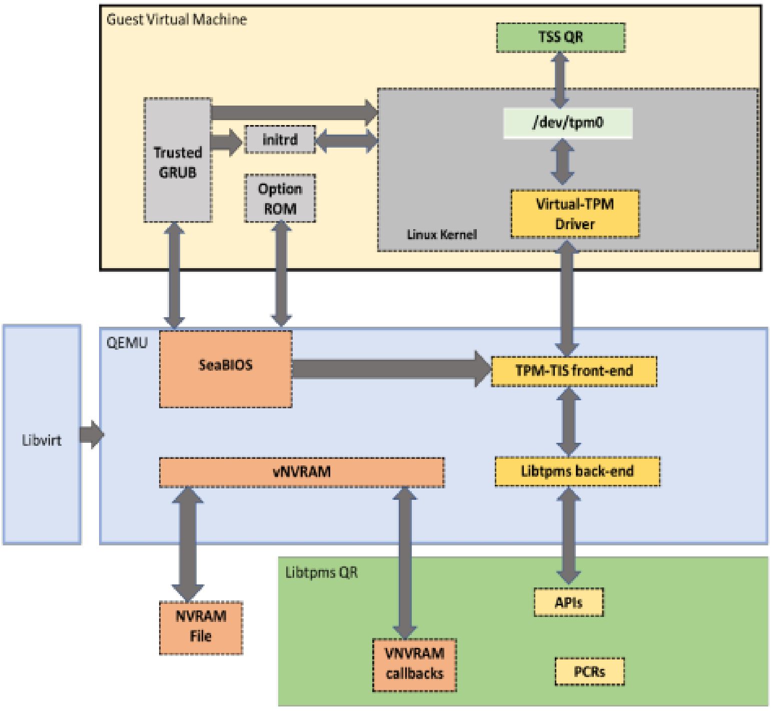


Figure 6: V-TPM Architecture

Within the QR V-TPM, the consortium tested and measured TPM commands of the following algorithms: Dilithium, Kyber, Rainbow, SPHINCS+, and BIKE.

Each of the algorithms listed above run and compile within the V-TPM environment. The same applies also for the newly added algorithms, namely Rainbow, SPHINCS+ and BIKE, which work within the V-TPM environment and can be compiled and executed within the V-TPM architecture.

3.2.2.1 Performance overview

The performance overview in the context of D6.6 is focused on the performance measurements taken for Rainbow, SPHINCS+ and BIKE QR algorithms, as the evaluation of the Kyber and Dilithium has been reported for the case of the SW-based implementation of the QR TPM in Section 3.2.1.2. Table 6 presents the execution timings of basic functions for the QR algorithms of the V-TPM. Based on evaluation results, Rainbow required considerable time for the key generation that the other two algorithms but performs better when it comes to the generation of the signature scheme and the signature verification in contrast to SPHINCS+. **Rainbow's key generation proved to be significantly slower, and this is mainly due to the keys used for Rainbow.** In fact, as can be seen in Table 7, the size of the generated key was significantly greater than the one generated for SPHINCS+. The memory usage of Rainbow, SPHINCS+ and BIKE QR algorithms, gathered by executing the specific commands for the V-TPM under Valgrind tool. More details on the performance of the aforementioned QR algorithms in the context of V-TPM can be found in D6.5.

Table 6: Execution timings for V-TPM based on evaluation results.

ms	Generate Key	Encapsulate	Decapsulate	Generate Signature Key	Verify Signature
BIKE	6	5	6	---	---
Rainbow	589	---	---	26	24
SPHINCS+	11	---	---	66	76

Table 7: Memory measurements for V-TPM-based experiments

kB	Key Generation	Signature Generation	Signature Verification	Encapsulation	Decapsulation
BIKE	---	---	---	1	1.015
Rainbow	3635.54	189.83	189.83	---	---
SPHINCS+	51.24	41.57	40.57	---	---

Overall, all the experiments conducted through the technical work packages of the FutureTPM project advocate that the V-TPM runs slower than the SW-TPM, but still with an acceptable overhead. This is due to the fact that the V-TPM is running within a VM on the host machine which causes some delay for each of the commands to be executed. In fact, the V-TPM is running a QEMU image that causes overhead and delay once the command has been sent.

3.2.2.2 Challenges on the Integration of Software TPM in Virtualized Environments

The integration of a software QR-TPM in virtualized environments proved to be more challenging than expected. Unfortunately, it cannot be easily plugged in, as a replacement to the current TPM 2.0, as all the software involved in the path between the QR-TPM and the application need to be modified. By taking as a starting point the QR version of libtpms, the library which implements all TPM commands, and the QR version of the TSS library, these are the software components that need to be modified:

- The wrapper of the libtpms library, that creates a backend suitable for use by the virtualization software (swtpm).
- The software in the host that emulates a real device for the virtual machine (QEMU).
- The virtual BIOS (SeaBIOS).
- The software responsible to receive and deliver TPM commands from/to the user space applications (the TPM driver in the Linux kernel).
- Third-party application libraries to do operations with QR algorithms outside the TPM (e.g., openssl).

There are two main causes why the QR-TPM cannot be easily plugged in: i) maximum size of the buffer, that should be large enough to contain QR keys, ii) TCG data structure definition changes, in particular the size field changed from 2 to 4 bytes.

Regarding the first issue, the main problem is that the buffer used to temporarily store the requests and the results from the TPM is fixed, usually 4096 bytes. That is, when integrating SPHINCS+, BIKE and Rainbow, the first issue we encountered was due to Rainbow having very large public keys. This size is not sufficient to contain for example a Dilithium key with mode set to 2 (TPM Load command size: 4152 bytes without key policy, 4246 bytes with key policy). Keys generated with higher modes have larger size. Whenever a larger key than the maximum size is used, the software returns an error, and the operation cannot be completed. Only in the case of QEMU this problem can be avoided, as the size of the buffer can be set dynamically by querying swtpm. In the long term, we hope that this approach, dynamically setting the buffer size, is also followed in other software such as the Linux kernel. However, for rapid prototyping we opted to change the TPM_BUFSIZE definition in the TPM driver of the Linux kernel and to perform similar changes in the other parts of the code whenever necessary.

For the second cause, unfortunately the only possible solution is to change the TCG data structure definition whenever the software itself interacts with the TPM and fills the buffer with the command. Otherwise, as expected, marshalling and unmarshalling functions return an error. An alternative solution could be to define new data structures and commands whenever it is necessary to send data larger than 65535 bytes. This would allow unmodified software to continue to work with the QR-TPM by specifying the conventional algorithms and to significantly minimize the changes to use the QR algorithms (use new algorithm IDs and new flags). The only functionality that could not be used unless existing software is modified is probably LDAA.

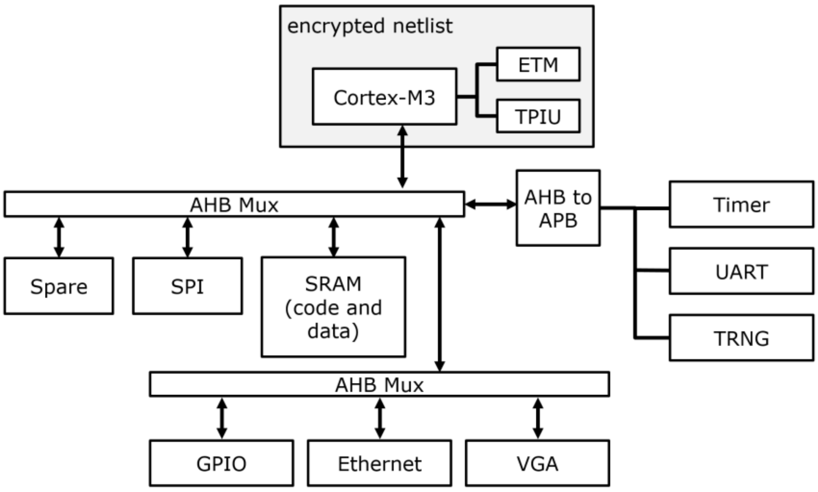


Figure 7: Simplified system overview on the MPS2+ DesignStart SoC

3.2.3 HW-based QR TPM Implementation

The HW-based QR TPM implementation was deployed on the system presented in Figure 7, which is based on an ARM Cortex-M3 CPU and SoC subsystem synthesized for an FPGA evaluation platform (ARM MPS2+). The aforementioned setup was selected due to the fact that the ARM Cortex-M3 CPU is similar to the SC300 CPU for secure elements. Notably, the SC300 combines the benefits of the industry standard Cortex-M3 processor with the proven security features of Arm SecurCore processors for embedded security applications. In addition, the adopted setup is more scalable, as such a FPGA architecture allows the extension with HW-based coprocessors and provides the necessary flexibility for its integration and testing in ecosystem comprising heterogeneous devices and embedded systems with various configuration setups and operating systems (as is the case in our envisioned use cases and applications domains).

In this context, the FPGA hosts and executes the operating system environment, which includes a scheduler and a lightweight TCP/IP stack & Ethernet driver in order to interface with external systems. This host environment includes a TPM application, which is based on IBM TPM2.0 SW simulator. It has been extended accordingly in order to include the quantum resistant primitives. More specifically, the HW QR TPM was extended with the functional implementation of Newhope and Bliss QR algorithms. Via the TCP/IP communication channel over the Ethernet interface, the HW-based QR TPM was able to execute the necessary TPM commands triggered by the TSS on the host machine, as can be seen in Figure 8.

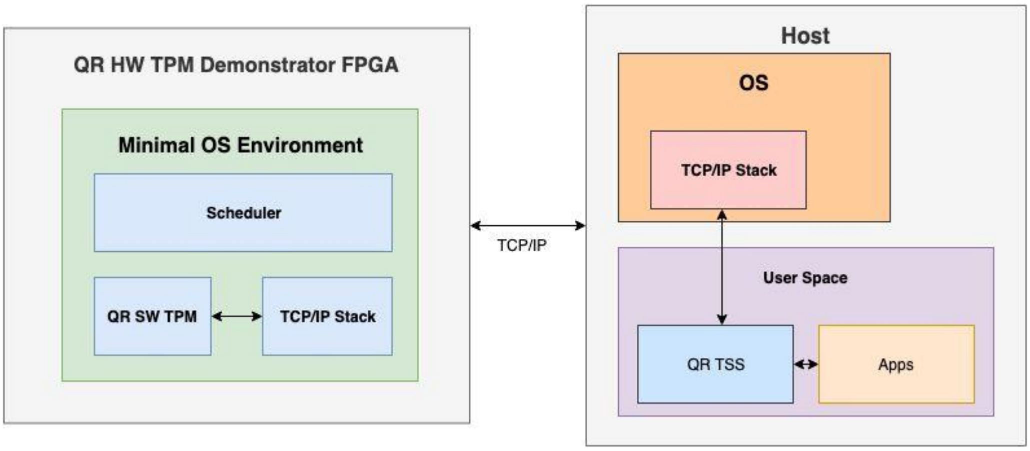


Figure 8: HW TPM System overview

In addition, the consortium implemented a coprocessor for the acceleration of quantum resistant cryptography in the context of D5.5 [26]. **The coprocessor is optimized for lattice-based primitives and can be used in an embedded design using a standard bus system.** It supports a large variety of lattice-based schemes and is designed to fit into the architecture of the HW TPM already outlined. It has been synthesized in a 65 nm standard cell library to obtain area data. Its performance is evaluated on the ARM MPS2+ FPGA system-on-chip (SoC) platform also used to realize the HW TPM.

The designed coprocessor is attached to the AHB bus of the ARM Cortex-M3 CPU and is thus easy to integrate into various designs. By using the so-called cached-NTT approach, the need for expensive memory in the coprocessor is reduced. Furthermore, this allows to realize the whole design in semi-custom logic without any hard macros or full-custom design elements. To enable easy integration into a cryptographic software, a specific driver was develop in order to interface with the coprocessor.

3.2.3.1 Performance Overview

The performance evaluation of the HW QR TPM was conducted in the context of the ePayment demonstrator. The demonstrator which has been designed and developed during the FutureTPM project is based on a refactored mobile application and brings into the picture TPM functionalities towards securing sensitive tokens, and facilitates a newly designed set of remote attestation and integrity verification enablers. That is, the demonstrator architecture was adapted to integrate the QR HW-based TPM variant, which is released on an FPGA-based board exposed by TCP/IP.

The evaluation of the demonstrator which is presented, in detail, in the context of D6.5 [19], required the interaction with the attached HW QR TPM towards the: (i) enforcement of produced security attestation policies (extracted by the FutureTPM Risk Assessment Engine) for the instantiation and execution of the newly developed **attestation protocols** namely, *Attestation by Quote* and *Attestation by Proof* (defined in D4.4 [17]), and (ii) execution of various QR cryptographic primitives ranging from traditional **digital signatures** to more advanced **key management operations** for enhancing the overall security posture of all involved actors and stakeholders in such an e-Payment ecosystem. In this context, and based on the defined user stories, the consortium evaluated the performance of the NewHope (Key Generation) and BLISS (Key Generation, Signature and Verification operations) QR algorithms. Table 8 depicts the performance timings taken for these basic operations of the aforementioned QR algorithms.

Table 8: Execution timings for QR HW-TPM based on evaluation results of D6.5. This the average of 100 executions.

ms	*Key Creation*	*Signature*	*Verify Signature*	*Encryption*	*Decryption*
NewHope	781,62	---	----	763.23	765.04
BLISS	40242.62	1535.49	601.12	---	---

NewHope Key Creation and Encryption/Decryption Operations: Regarding the performance of the NewHope primitives, as can be seen in Table 8, the key generation (CC_Create command) requires on average 781,62 ms to complete. After creating the NewHope key pair, the encryption and decryption operations are performed using the CC_NEWHOPE_Enc and CC_NEWHOPE_Dec, TPM commands respectively. These commands have been tested in the context of INDEV.AU.3 user story for the encryption and decryption of a transactions' database. The aforementioned operations required almost similar time to complete, achieving 763.23 ms and 765.04 for the encryption and decryption, respectively.

BLISS Key Creation and Signing Operations: The BLISS signature scheme was used for creating the necessary Attestation Key used as the trust anchor during the Attestation-by-Quote and -by-Proof approaches. Thus, the CC_Create command triggers the process of the key creation of BLISS symmetric private key. As can be seen in Table 8, **this operation needs a considerable amount of time to complete.** More specifically, after 100 execution instances, the extracted average time converged to around 40.2 secs. However, we have to highlight that the distribution of the monitored timings also demonstrated a rather unusual high standard deviation. This notable behaviour is depicted in Table 9 where it can be seen that BLISS has a non-deterministic mode of operation that results in such high deviations in the monitored performance. This is further ratified by the observed range (Max-Min) of 255.6 seconds, while the standard deviation of the results is 42.63 seconds. Figure 9 reveals the distributional characteristics of these results. The Median is placed at around 30 secs, the 1st and 2nd quartiles being rather concise, but the 3rd and 4th being rather expansive, and thus, affecting the average performance to converge approximately to 40 secs. The BLISS implementation which was integrated in the HW QR TPM can be found in the GALACTICS repository [33]. Given this implementation one can see that during key generation there are multiple steps, where randomness of the primitives may be rejected, and the generation process is initiated again. That is, given this implementation approach, the deviation in the performance of the BLISS key generation is justified, as the process tries to maximise the randomness and several iterations may occur to achieve this goal.

The BLISS scheme is used to derive the attestation keys to facilitate the attestation schemes. In this regard, the signing and signature verification operations are important to evaluate the overall efficiency. Thus, the average of 1535.49ms is required for the signing process and 601.12 ms for the signature verification.

An extensive set of performance timings of the QR algorithms in the context of the HW QR TPM can be found in D6.5 [19].

Table 9: CC_Create command statistics over 100 experimental results for the BLISS key creation.

CC_Create (BLISS) statistics	
Max	263.54 seconds
Min	7.94 seconds
Range	255.6 seconds
St. Deviation	42.63 seconds
Coefficient	0.9822

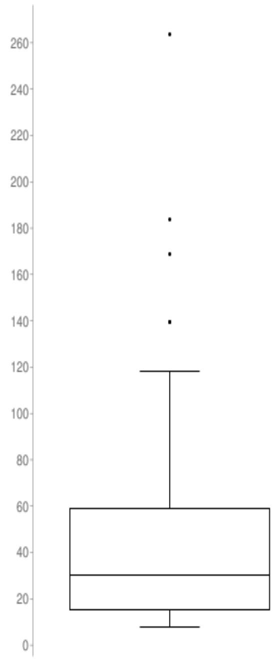


Figure 9: Boxplot of CC_Create of BLISS key pair over 100 executions.

3.2.3.2 *Discussion & Critique*

This section offers a discussion of the results, learnings, and notable findings that one needs to consider for the design of an actual hardware QR TPM. It must be noted that the larger public-keys and large internal data structures that QR algorithms suggest can pose significant challenges for embedded devices.

For instance, **the experimentation with Dilithium-IV required to increase the capacity of I/O buffers, as the biggest signature for Dilithium-IV is 3,3 Kbyte.** In the same direction, the internal RAM consumption is critical, as Dilithium-IV requires 40 to 80 Kbytes of RAM. These outcomes need to be considered especially when it comes to the integration of a QR TPM in resource constraint devices. As an example, if one considers the use case environments of the FutureTPM demonstrators, where there are heterogeneous types of devices with limited resources, the requirement to increase the capacity of I/O buffers and RAM poses a challenge that need to be taken under consideration. On top the above, one needs to consider also that the RAM consumption can be increased due to the consideration of attack countermeasures to be deployed.

Regarding the performance analysis of the HW QR TPM, based on the experiments conducted through the technical work packages and considering the evaluation results of the Secure Mobile Wallet and payments use case in D6.5 [19], the performance was satisfactory, and the demonstrator met almost all the quantitative KPIs of the project. In addition, no significant overhead introduced by TPM to support the QR algorithms. It must be noted that the demonstrator met the KPIs based on the HW QR TPM implementation on an FPGA board, which was not optimized for the ARM architecture. In addition, the FPGA includes a simple scheduler which consumes ~50% of the available CPU time. Hence, we need to highlight that the performance was as expected, since the deployment was performed on a Cortex-M3 CPU-equipped board with a time-slice-based OS.

In an effort to enhance the performance of the HW QR TPM, the consortium also worked towards HW acceleration supported by a co-processor optimized for lattice-based primitives (D5.5 [26]). This effort revealed that **HW-acceleration is less straightforward than for ECC/RSA** due to more diverse parameters and operations (poly/matrix arithmetic, sampling, decoding) for the case of the QR algorithms. However, further optimizations in the implementation aspects of the algorithms (such as NewHope on which the consortium worked for IFAT’s Cortex-M3 processor) would be able to better leverage the capabilities of such crypto acceleration techniques and underlying processors. When the share of other operation on the overall cycle count is decreased, the impact of the NTT acceleration becomes more visible. In addition, it would be interesting to evaluate whether the NTT co-processor could be used to accelerate non-NTT schemes like Saber and NTRUPrime (see also [35] for a first analysis.

Chapter 4 Provable Security Modelling and Formal Verification of QR TPM

Key Takeaways on Provable Security Modelling and Analysis of QR TPM

- The security modeling and formal verification of the TPM was based on a "bottom-up" approach, in which the focus is on modelling the core TPM functionalities towards building chains of trust (instead of considering the TPM as a whole).
- Development of a *trusted platform commands abstraction model* consisting of a specific set of formally-specified primitives sufficient to implement the core TPM functionalities beyond the core crypto operations. Such an abstraction modelling can enable the reasoning about and comparing different TPM services under various adversarial models and for different security guarantees, excluding any possible implications from the leveraged cryptographic primitives.
- This break-down of TPM ideal functionalities and services allows for a more effective verification process towards building a global picture of the entire TPM platform security modelling as a Root-Of-Trust. These models are designed to be modular and amenable to extension by the community.
- The consortium developed a new property-based DAA model, which is a combination of a traditional game-based model and a Universal Composability (UC) model, to verify the security of the designed Lattice-based Direct Anonymous Attestation (LDAA) schemes.
- Formalized the notion of secure remote attestation towards trust aware service graph chains (in the context of the envisioned Device Management use case) and presented Tamarin security proofs showing that our models satisfy the three key security properties that entail secure remote attestation and execution: integrity, confidentiality, and secure measurement.

4.1 Assessment of the Security Modelling for the Trusted Platform Module as a Root-of-Trust

The core objective of WP3, is the security modelling of the TPM and the formal verification of its security properties. In D3.1 [28] and D3.2 [29] we described a number of research challenges for the modelling of the TPM functionalities, and for capturing its usage in security protocols and the interaction of different parties with the TPM. In the context of the three use cases that are studied in this project, but also in most TPM-based applications in the literature, it is usually the case that only a small and specific subset of TPM functionalities is utilized. This, in combination with the high complexity of the TPM and its interaction with the outside world, raised concerns on whether an attempt to develop a security model for the TPM as a whole that captures all TPM functionalities would be feasible within the timeframe of this project. In order to avoid such a risk, we revised the research plan for the security modelling of the TPM.

Our refined plan, as it is described in D3.3 [30], was to instead of looking at the TPM as whole, focus on the modelling of the subset of its core functionalities, namely the functionalities that are a common reference in the project use cases, as well as in the vast majority of TPM-based protocols in the literature. As such, the most notable TPM functionalities are the process of creating a TPM key and loading it into the TPM, the Enhanced Authorisation (EA) mechanism, which is a new feature in TPM2.0, the management of PCRs, the use of NV memory for secure storage and remote attestation. **Remote attestation is perhaps the most popular functionality of the TPM and it can be accomplished either by the built-in Direct Anonymous Attestation (DAA) protocol, or through the usage of a Privacy Certification Authority (PCA) as is the case of the IBM remote attestation protocol** [36], which is considered in one of the use cases.

The modelling of the core TPM functionalities is essentially equivalent to the modelling of the most significant and frequently used TPM commands. Our approach here is to model the semantics of these TPM commands in such a way that we exclude the cryptographic operations used internally by the TPM (e.g., hash functions and public key encryption), and replace these operations with non-cryptographic approaches (*trusted platform commands abstraction model*). We note that we retain the cryptographic operations that are part of the application itself, excluding only internal TPM cryptography for internal consumption. Examples of non-cryptographic mechanisms include a combination of equational theory, which is a common technique in verification tools, such as SAPiC, Tamarin or ProVerif, the use of a Trusted Third Party (TTP) or involving private channels to model the communication between parties in a secure manner. **We refer to the modelling of a TPM command in this way as an "idealized functionality" of the command.** When using these ideal functionalities, the assumption needs to be made, that the TPM is "almost as secure" as our ideal functionalities.

In D3.3 [30], we presented an example of these ideal functionalities, specific to the process of **TPM Key Creation and Loading**. This is one of the most fundamental procedures in any protocol that involves the use of a TPM. The TPM key is linked with a policy, which is used in the context of the EA mechanism for authorising the key loading process. Motivated from the use cases of the project, we used a PCR policy for this purpose. The next step would be to apply this model in more complex scenarios and particularly extend its use in the context of the use cases of the project. We elaborate on this task exclusively and extensively in deliverables D3.4 [6] and D3.5 [7].

4.1.1 Modelling Approach

In order to apply our modelling techniques in practice, we first need to choose a use case that will serve as a reference example and will be representative of most of the core TPM functionalities. For these purposes, we chose to start with project's Use Case #3: Device Management, which is led by Huawei partner. The reason for choosing this use case as a starting point is because it includes the aforementioned TPM functionalities, namely the creation of TPM keys, EA authorization, PCR management, policy sessions and remote attestation. The device management use case describes a network infrastructure which is composed of a (set) of routers, a Remote Attestation (RA) server and a Network Management System (NMS), where each router is equipped with a TPM. The goal of the use case is that the NMS is able to determine a routing policy based not only on network topology assumptions, but also on the trust assumptions of the nodes that compose the network. To achieve this, each router that is part of the network must be able first, to securely join the network (enrolment process) which is managed by the NMS, and second, to be able to attest its status upon request by the NMS. This second phase is carried on through the establishment a secure TLS communication channel with the NMS, and subsequent sending of the attestation report. From a high-level point of view, these tasks can be decomposed into the following main actions below (see the use case description in [10] for the complete details):

1. The Router creates an Attestation Key (AK) using the TPM, which is certified by the RA Server. The trust assumption relies on the Endorsement Key (EK), which is certified by the TPM manufacturer.
2. The Router creates a TLS signing key using the TPM, which is also certified by the RA Server upon reception of the Certificate Signing Request (CSR). The TLS key is also signed by the previously certified AK, and included in an extension of the CSR.
3. The Router uses this TLS key in order to securely communicate with the NMS, and it forwards a TPM quote (signed with the AK) providing evidence of its integrity.

Once a specific TPM-based service is selected, we can create a security model that captures the actions of the TPM and the interactions of the involved parties, by following the next modelling steps. We note that the modelling approach we describe here is not necessarily specific to this use case, but it can be extended to any TPM-based application.

Step 1: Determine the TPM commands. The first step in our modelling approach is to identify the TPM commands that are used in the chosen use case that will be modelled. The important part in this step is to fully realise how these commands operate and the exact actions of the TPM commands that need to be modelled when each command is executed. In order to do this, we presented an abstract description of the required functionalities of the TPM commands considered. Concretely, this abstract description is a high-level explanation of how each TPM command works, based on the TPM specification manual [37], by avoiding some technical details that are irrelevant to the modelling. In addition to having a better view of the TPM commands, such an abstraction will allow us to model each command in such a way that our "ideal functionalities" are as close as possible to the TPM operation in real life.

In the device management use case, the TPM commands that are needed, are those related to the creation of a TPM key: **`TPM2_StartAuthSession`**, **`TPM2_PCRExtend`**, **`TPM2_PolicyPCR`**, **`TPM2_Create`**. The abstraction of these commands is presented in D3.3 [32]. In addition, for the remaining parts of the use case, related to the certification of keys, we also require the commands **`TPM2_MakeCredential/TPM2_ActivateCredential`**, for creating and activating a credential, **`TPM2_Certify`**, which ensures that an object is indeed created by the TPM and the crypto-related commands **`TPM2_Sign`**/**`TPM2_VerifySignature`** and **`TPM2_PK_Encrypt`**/**`TPM2_PK_Decrypt`**, where the latter two refer to public key encryption and decryption respectively. The abstraction of these TPM commands can be found in Deliverable D3.4 [31]. Finally, the command **`TPM2_Quote`** is required in the third step of the use case, namely the attestation of the router integrity after the creation of the TLS communication channel between the Router and the NMS. The abstraction of this command is given in Deliverable D3.5 [32].

***Step 2: Define the ideal functionality for each command*.** Based on the abstractions that we have created, we define the ideal functionality for each TPM command. This is a model of the command which excludes the cryptographic operations that are carried out by the TPM and replaces them with non-cryptographic mechanisms. Our ideal functionalities for the TPM commands in the device management use case are described in Deliverables D3.3 [30] and D3.4 [31]. We also note here that when modelling these commands, we need to have a specific symbolic verification tool in mind. We have chosen the Stateful Applied Pi Calculus (SAPiC) tool for this purpose. Our motivation for using this specific tool is that SAPiC is based on the Tamarin tool, which can be used to verify security properties under the Dolev-Yao model [38], with respect to an unbounded number of sessions. It is also because several TPM-related examples in the literature, such as [40][41], that base their security analysis in SAPiC/Tamarin modelling. Most importantly, SAPiC is one of the few tools available that has support for non-monotonic global mutable state, a feature that is necessary to model TPM-based protocols, as a TPM is essentially a stateful device.

***Step 3: Model the remaining components*.** The ideal functionalities correspond to the model for the TPM. We then need to model the remaining parties of the protocol, as well as their interaction with the TPM. In the device management use case, these parties are the Router, the RA Server and the NMS. Specifically, it is the Router that interacts with the TPM, in order to create the AK. The certification of the AK is done with the IBM remote attestation protocol [36], which is executed between the Router, the TPM and the RA Server. We have modelled this part in Deliverable D3.4 [31]. Then the Router creates a TLS signing key, using the TPM, which is also certified by the RA Server, using the AK certificate that was previously created through the IBM remote attestation protocol. A recap of the AK certification model is also presented in D3.5 [7]. The detailed model for the TLS key creation and its certification is presented in Deliverable D3.5 [32]. The last part, as it is described in the beginning of this section, is the actual TLS communication of the Router with the NMS. The modelling of this part is extensively studied in Deliverable D3.5 [32]. In order to simplify our model, the above three steps in the device management use case are modelled independently. Our model also abstracts away the communication of the NMS and RA server, as well as the database operations executed in their side, as these are considered trusted operations and are out of the scope of the threat model.

***Step 4: Formal verification of security properties*.** The final step in our modelling approach is to define the security properties we need to capture and formally verify them using automated proofs. This verification is done by expressing the desired security properties as Tamarin first-order formulas encoded in lemmas whose syntax is presented in the Tamarin manual [12]. The security properties we aim to verify are:

- *Sanity checks:* show that the model executes (reaches) all possible branches.
- *Availability of keys at honest processes:* verify that all honest parties have access to the key material required.
- *Key freshness and secrecy:* prove that the created keys during the execution of the model are fresh and not available to the adversary.
- *Authentication:* prove the agreement property as described in Lowe's hierarchy [42]. Depending on the case, this property can be regarded as mutual and/or injective.
- *Transfer of information as generated:* ensure that cryptographic material (e.g. AK and TLS certificates, or TPM quotes) are received by the destination process as generated by the process of origin.
- *No reuse of key:* ensure that specific cryptographic material is fresh and only used once.

All the above properties have been considered for all three models; AK certification, TLS certification and TLS communication establishment. A detailed analysis for modelling the above security properties, as well as their formal verification is given in Deliverable D3.5 [32] .

4.1.2 Challenges

When working with formal verification tools, there are several difficulties that one needs to take into account. As the verification in the symbolic model is undecidable for an unbounded number of sessions and term length, a common drawback when one proves security in this model is that certain lemmas either require too many steps in order to be proved, or they are not proved at all. In other words, there is no guarantee that the tool will terminate proving a certain security property, even if the model is specified correctly. In addition, we require to model non-monotonic global state, which introduces an additional difficulty for the tool. As we discussed earlier, we used SAPiC in our modelling, which is based on Tamarin. It translates protocols specified in the stateful applied pi-calculus into a set of rewrite rules. We were able to prove most of the desired security properties, however, some of our lemmas required a lot of steps or manual intervention in order to be verified. The kind of manual intervention required to address non-termination was in the form of 1) providing manual "source lemmas", which are required to remove open chains (i.e., to guide the tool to find the source of certain messages), 2) simplify and abstract away as much as possible implementation details that do not play any role in proving security properties, 3) reduce as much as possible the usage of global state, and 4) reduce as much as possible the lock/unlocking mechanism, as it is a very expensive operation from the point of view of verification.

Another challenge that we have faced and is worth to mention is the communication between the Router and the TPM. If we want to model these two entities as separate processes, this communication channel should be treated in a special way for an outsider adversary (i.e., an adversary that has not compromised the router). As the Dolev-Yao model [38] suggests, the adversary is able to interfere in any way in the communication between the Router, the RA and NMS Server (which are regarded as a single process), however, the communication between the Router and the TPM is trusted and the TPM output should not revealed to, or tampered by the adversary. Hence, we had to explore some alternatives to achieve this goal. Theoretically, this secure communication between the Router and the TPM can be modelled using private channels, however, as we observed, this further complicates our model: *private channels are regarded as synchronous channels by the tool and this makes the translation even more complicated.* Another solution would be to use global states. That is, whenever the Router wants to provide an input to the TPM (equivalently the TPM outputs a value to the Router), it inserts that input in a global variable. On the other hand, in order for the TPM to receive an input from the Router (equivalently the Router receives an input from the TPM), it reads the value that is stored in the global variable. Although this solution might work better in some cases, as it suppresses the "synchronous" behaviour of private channels, it also introduces more persistent state management, and it has not worked well for our purposes either. Our final solution for limiting the capabilities of the adversary in this private channel was a combination of the usage of restrictions (i.e., enforcement of properties in traces of the protocol) and the usage of an advanced SAPiC feature to allow direct manipulation of state facts. The semantics of the resulting proposal is not a traditional secure channel, but a channel in which the adversary has no access to some messages, and it can only forward (without modifying or replaying) some messages. Therefore, if any property holds for a channel of this class, it will certainly hold for a private channel.

Apart from the aforementioned challenges, it needs to be stated that the security modelling of the TPM and the formal verification of its security properties was a challenging task per se. The initial vision of the FutureTPM project was to develop a security model for the TPM as a whole that captures all TPM functionalities. However, the consortium decided that such an approach would not be possible to complete in the time frame of the project. That is, we addressed this problem by breaking down the verification process and performing the security modelling of the TPM and the formal verification to specific TPM functionalities. In fact, the difficulty of this endeavour was also confirmed by the Advisory Board members, and our revised plan to break down the verification process was endorsed by the board members as the appropriate approach to tackle this issue.

4.2 UC Proof for LDAA

The security of our designed two Lattice-based Direct Anonymous Attestation (LDAA) schemes, introduced in D2.1 [12], D2.2 [27] and D2.3 [20], are proved under a Universal Composability (UC) model. In this model, an environment, *E*, passes inputs and outputs to the protocol parties. The network is controlled by an adversary, *A*, that may communicate freely with *E*. There are two worlds in this model: an ideal world and a real world. In the ideal world, the parties forward their inputs to the ideal functionality, *F*, which then (internally) performs the defined task and creates outputs that the parties forward to *E*. In the real world, a real-world protocol, *Π*, is said to securely realize a functionality, *F*, if the real world is indistinguishable from the ideal world, meaning for every adversary performing an attack in the real world, there is an ideal world adversary (often called simulator), *S*, that performs the same attack in the ideal world. More precisely, a protocol *Π* is secure if for every adversary *A*, there exists a simulator *S* such that no environment *E* can distinguish executing the real world with *Π* and *A*, and executing the ideal world with *F* and *S*.

The security proof of each LDAA scheme includes a sequence of games. Start from the first game, which is the real-world protocol. Then gradually introduce different functionalities in each other game; every time we prove that from the environment *E*'s point of view, this new game is indistinguishable from the previous game. Eventually, the last game includes the designed ideal functionality of a DAA scheme. In the proof of each game, if the adversity *A* has a non-negligible probability to break our LDAA scheme, then the simulator *S* can solve a lattice-based problem, which is assumed to be hard to solve. Therefore, when such a sequence of the games reaches to the end, we have proved that the designed LDAA scheme holds the desired properties that are interpreted under the ideal functionality, *F*.

More information and discussions of the DAA security modelling have been given in D2.1[12], D2.3 [20], D3.2 [29] and D3.3 [30].

4.3 Future Work and Extensions

As we mentioned previously in Section 4.1, our revised plan was to break down the verification process and focus our modelling on a subset of core functionalities of the TPM. In this direction, we identified the most prominent functionalities and services that need to be modelled. That is, we chose as our starting point the remote attestation, as the most important service in a variety of application domains, and in fact it is also a common denominator in all use cases of the FutureTPM project. In order to formally verify more functions, we studied the device management use case which had the wider spectrum of engaged functionalities among the use cases, including the generation and management of attestation and TLS keys. Thus, we managed to formally verify some of the most significant security properties and functionalities provided by the TPM, namely the creation of TPM keys, the EA mechanism and PCR management. We argue that our approach to model the core

TPM commands instead of looking at the TPM as a whole, can be followed in order to model the two additional use cases of the project, namely Use Case #1: Secure Mobile Wallet and Payments and Use Case #2: Activity Tracking. Further, we argue that our decision to start our modelling with the device management use case was correct, since it contains the most commonly used TPM commands in most TPM-based applications. This is justified in the following paragraph.

In particular, we argue that the idealized functionalities, considered as part of the trusted platform command abstraction model, are a common reference point in most TPM-based services. Therefore, since the intuition behind such an abstraction is the definition of a generic model that can serve as a specification of primitives for TPM operations, we believe that the decomposition of additional features and functionalities - needed by other application domains and environments - would require only minor refinements and the modelling of a small set of extra TPM commands. This is mainly due to the fact that most of nowadays TPM-based applications are using the already (FutureTPM) considered algorithms and protocols in an attempt to create trust aware service graph chains: namely, the DAA protocol, EA mechanism, and PCR management. **This can act as evidence on the generality and applicability of the produced models to be considered as an extensible verification methodology for enabling rigorous reasoning about the security properties of Future TPMs.**

4.3.1 Extending our Models to Other Application Domains

Considering the work performed on the security modelling of TPM-based functionalities, a logical future step is the instantiation of the formal models of the remote attestation to other application domains. As have been stated, the remote attestation is a prominent TPM-based service that contributes to the security posture of various application domains. That is, this future action is highlighted as the first step towards this direction. On top of the remote attestation, it is vital to formally verify additional key services, such as EA mechanism, key storage, management of the platform configuration register in these domains, but also to plan the same strategy for the application domains of the rest of the use cases of the FutureTPM project.

Use case #1: Secure mobile wallet and payments: The main TPM-based functionality in this use case is the *sealing* process. This procedure involves the generation of a random password and the creation of a TPM key that is linked with specific PCR values through a PCR policy session. The authorization value (*authValue*) of the TPM key is set as the random password that is previously generated by the TPM. This implies that the password is sealed into specific PCR and can be obtained using the command **`TPM2_Unseal`**. The process of creating the TPM key is exactly the same with the creation of the TLS key in the device management use case, the only thing that differs is the usage of the key. Sketching the security model for this use case, we see that the ideal functionalities of the TPM commands we have presented can be easily applied in this use case, with the addition of the ideal functionality of the command **`TPM2_Unseal.`**

Use case #2: Activity Tracking: At the core of this use case is the execution of the DAA protocol between the Platform (S5PersonalTracker or S5DataAnalysis and TPM) the Issuer and the Verifier (S5Tracker Analytics Engine). The TPM commands that are required for the execution of the DAA protocol are exactly the same ones that we have modelled for the device management use case. There are two additional TPM commands that are needed, **`TPM2_Hash`** and **`TPM2_Commit`**, where both perform cryptographic operations on behalf of the TPM. This means that for modelling purposes they can be replaced with non-cryptographic approaches. In addition, the DAA key that is created by the TPM is also linked with a PCR policy. Therefore, our modelling of the ideal functionalities in the context of device management can be easily applied in this use case as well.

We further argue that this “bottom-up” approach for the security modelling of the TPM, that we described in Section 4.1.1, can also be followed for modelling various complex TPM-based scenarios in the literature. Nonetheless, the ultimate goal is to collect all the pieces of the puzzle, so as to combine all this models on the verification of services for delivering a holistic and unified trust model that covers the whole spectrum of the TPM-based services. In addition, Potential extensions of our model could involve the use of different policies instead of, or in conjunction with the PCR policy instance, the modelling of key hierarchies and key management in the TPM, or the modelling of scenarios that make use of the TPM's NV memory. An interesting scenario is to also consider applications with multiple TPMs interacting together (e.g., creating a TPM key and embedding it in another TPM). More concretely and in the context of trusted computing, we believe that our device management security model can be viewed as the basis for modelling various applications and protocols involving alternative cryptographic hardware devices in general.

4.3.2 Future Actions towards Formalizing LDAA

The Universal Composability (UC) model is one method to prove a complex protocol, like a LDAA scheme. **Our work on proving two LDAA schemes discussed in Section 4.2 has demonstrated that this model is successful.** However, the UC model is not the only choice for proving a DAA scheme. In the literature, researchers have used a game-based model, which is also called a property-based model, or a simulation-based model to prove DAA schemes. In those early works, security of the DAA schemes is based on the factorization problem or the discrete logarithm problem. **In the FutureTPM project, our consortium participants have developed a new property-based DAA model, which is a combination of a traditional game-based model and a UC model.** This new model has been introduced in D3.3 [30]. While this model has not been directly instantiated for our new L-DAA algorithm, it has been extensively studied and designed for the traditional DAA protocol. As both variants target the same security and privacy properties (differentiating, though, the crypto primitives leveraged) of user-controlled anonymity, unlinkability and platform authentication, this hybrid model is envisioned to also offer the same level of security proofs in the lattice-base domain. However, this is something that will be evaluated as a future work for the latest version of the new L-DAA algorithm

As it was discussed in the earlier part of this document, the existing LDAA schemes are not as efficient as we aim for and there is still room for improvement. In the future research, we will continue to work on designing new quantum-resistant DAA schemes, which should have better performance, and we will also try to prove a new quantum-resistant DAA scheme under the new property-based DAA model.

In addition, as a future action we aim to **evaluate the applicability of the method used in the new property-based DAA model, which is a combination of a traditional game-based model and a UC model, to other QR algorithms with similar security and privacy considerations as the DAA**. In this way, we aim to enhance the set of formal methods of the domain by introducing this hybrid modelling approach that could benefit similar future actions by the community. Our aim is to contribute with a methodology which eventually will result to the design of more robust cryptographic protocols with strong guarantees on their security properties. In this context, such an approach could benefit the community by forming a new specification that could benefit future modelling actions.

Chapter 5 Summary and Conclusion

Summing up, this deliverable critically appraises the technical developments of the project, highlights the lessons learnt, with regards to the implementation, integration, operation and execution of the demonstrators, while it provides adoption guidelines when it comes to the integration of QR algorithms in a Future TPM.

Considering the value propositions of the project, we provided an overview of the challenges and we critically discussed the evaluation results of the project's outcomes, considering also the impact assessment highlights of the demonstrators. Our analysis and critical appraisal were based on the following pillars:

- The FutureTPM framework and its building blocks
- The design and implementation of the generation QR Trusted Platform Module
- The provable security modelling and analysis of the TPM

The development of FutureTPM framework was driven by the emerging challenges of decentralised system architectures that pose strong security, privacy and trust requirements. In this direction, the **FutureTPM project provides a framework that can defend the security posture of future hyperconnected ecosystems through the integration of decentralised roots of trust that can resist against powerful adversaries in the post quantum era.** Toward this vision, the consortium developed a reactive run-time risk assessment and mitigation framework to ensure security of use cases in the face of emerging threats and vulnerabilities.

Having the QR TPM as the trust anchor, the risk assessment framework serves the project's vision through the development of the FutureTPM Control Flow Property-based Attestation Toolkit to achieve Integrity Verification and runtime operational assurance of the assessed systems. To do so, the framework introduced two new attestation schemes, namely the **Attestation by Quote** and **Attestation by Proof**, that tackle the documented limitations of the literature and address privacy concerns when it comes to the disclosure of identifiable characteristics about the attested platforms. The remote attestation schemes are complemented by the FutureTPM Secure Tracing & Trust Evidence Collection that capitalises on the dynamic multilevel tracing techniques using eBPFs and Intel PT in order to advance the state of the art of the control flow attestation solutions.

The aforementioned artefacts work in synergy to offer a holistic Risk Assessment framework capable of providing vulnerability analysis and policy enforcement during both design- and run-time. These offerings positively affect the application domains of the FutureTPM demonstrators. The impact assessment on behalf the demonstrators advocate that FutureTPM can offer high security and trust guarantees at the device, data and software system level and meet the need of earning end-users' trust through the use of the various security measures and controls.

As aforementioned, TPM is the heart of the project and our research and innovation actions for the development of a QR trust anchor have generated a set of artefacts and evaluation results that were critically appraised in the context of this deliverable. In this context, based on the QR primitives and algorithms investigated and integrated in the QR TPM variants, the consortium offered recommendations on cryptographic schemes that meet the security criteria posed in the PQ era. With respect to the asymmetric primitives, all four schemes (Kyber, BIKE, Dilithium, and SPHINCS+) are finalists of the ongoing NIST competition and standardization effort. The consortium also proposes five asymmetric primitives as secondary back-up choices (NewHope, NTTRU, BLISS, Rainbow, and Picnic.

Based on the analysis on the QR algorithms, we provided adoption guidelines for the design of the three variants of the QR TPM, namely the software, the virtual and the hardware QR TPM. The deliverable offered a thorough discussion on the implementation and integration challenges and highlighted the required TPM architecture specifications that need to be revised for the adoption of QR algorithms considering also the performance measurements and memory requirements collected from the lab testing.

Finally, the deliverable highlighted the challenges that the consortium faced towards the security modelling and formal verification of the TPM and provided adoption guidelines for future actions in this direction. The consortium followed a revised plan on dividing the modelling to individual functionalities of the TPM instead of performing the formal verification to the TPM as a whole. In addition, for the formal verification of DAA, which is a complex protocol at hand, the consortium offered a combination of a traditional game-based model and a Universal Composability model for the formal security modelling. Our vision is to extend this modelling and consider more algorithms that have similar security and privacy consideration such as DAA.

Overall, research on quantum computers has drawn enormous attention from the cryptographic community, government and industry. If, as predicted, a large-scale quantum computer becomes a reality within the next 15 years or so, existing public-key algorithms used in TPMs will be open to attack. The TPM industry faces the challenge of providing a smooth transition to Quantum-Resistant (QR) cryptography. A significant challenge to overcome in supporting this transition is the fact that TPMs, and other hardware devices that support security, often have very limited resources in terms of power, storage and computing speed. **The FutureTPM project has already provided a proof of concept implementation of all QR TPM variants (software-, hardware, and virtual-based) that already enabled the identification of various constraints that need to be resolved when integrating QR crypto primitives into the TPM, e.g., the size of the non-volatile memory provided by the TPM to hold the keys, the size of the buffer used to send parameter data to the TPM, etc.** This validated the project's initial vision towards the provision of QR decentralized roots of trust as an enabler to establish CPS communities of trust.

Under this perspective of quantum safe computing, FutureTPM outcomes represent an invaluable milestone to the edge of current trusted computing technological limitations and to offer a radically new class of services well beyond the state of the art. By supporting multiple security and trust-oriented deployments, this milestone was achieved through FutureTPM's: (i) personalized cybersecurity functions in heterogeneous SoS, (ii) enhanced operational assurance and detection capabilities by defining trust zones (that can be dynamically updated) comprised of distinct pockets of infrastructure where resources operate at the same trust level and similar safety-critical functionality, thus, minimizing the number of allowed pathways and limiting the potential for malicious threats to affect safety-critical applications, and (iii) automated orchestration of advanced security deployments in an easy and affordable way. The compiled architecture has the potential to provide a fully operational trusted service graph chain able to host all stakeholders of a SoS ecosystem aiming to form the first co-ordinated effort for a distributed (QR) safety-critical SoS solution.

Chapter 6 List of Abbreviations

Abbreviation	Translation
EA	Enhanced Authorisation
AES	Advanced Encryption Standard
AK	Attestation Key
BBA	Binary-Based Attestation
BIKE	Bit Flipping Key Encapsulation
BLISS	Bimodal Lattice Signature Scheme
CFB	Ciphertext Feedback
CIV	Comprehensive Integrity Verification
CPU	Central Processing Unit
CSR	Certificate Signing Request
DAA	Direct Anonymous Attestation
DSA	Digital Signature Algorithm
EA	Enhanced Authorization
eBPFs	enhanced Berkeley Packet Filters
ECC	Elliptic Curve Cryptography
FPGA	Field Programmable Gate Array
HMAC	Hash-based MAC
ICT	Information and Communication Technologies
IMA	Integrity Measurement Architecture
IntelPT	Intel Processor Trace
IoT	Internet of Things
KEX	Key Exchange
LDAA	Lattice-based Direct Anonymous Attestation
MAC	Message Authentication Code
NIST	National Institute of Standards and Technology
NMS	Network Management System
NTT	Number-Theoretic Transform
PBA	Property-Based Attestation
PCA	Privacy Certification Authority
PCRs	Platform Configuration Registers
PKE	Public Key Encryption
PQ	Post-Quantum
PQC	Post-Quantum Cryptography
QR	Quantum Resistant
RNG	Random Number Generator
RSA	Rivest–Shamir–Adleman
SAPiC	Stateful Applied Pi Calculus
SHA	Secure Hash Algorithm
SoC	System-on-chip
TC	Trusted Components
TCB	Trusted Computing Base
TCG	Trusted Computing Group
TCTI	Command Transmission Interface
TLS	Transport Layer Security
TPM	Trusted Plaform Module
TSS	TPM Software Stack
TTP	Trusted Third Party
U2F	Universal 2nd Factor
UC	Universal Composability
XOF	Extendable-Output Function
XOR	Exclusive OR

Chapter 7 Bibliography

[1] El Kassem, N., Chen, L., El Bansarkhani, R., El Kaafarani, A., Camenisch, J., Hough, P., Martins, P., & Sousa, L. (2019). More efficient, provably-secure direct anonymous attestation from lattices. Future Generation Computer Systems, 99, 425–458. https://doi.org/10.1016/j.future.2019.04.036

[2] Chen, L., El Kassem, N., Lehmann, A., & Lyubashevsky, V. (2019). A Framework for Efficient Lattice-Based DAA. CYSARM 2019 - Proceedings of the 1st ACM Workshop on Workshop on Cyber-Security Arms Race, 23–34. https://doi.org/10.1145/3338511.3357349

[3] TCG: TCG Guidance for Securing Network Equipment Using TCG Technology Version 1.0 Revision 29 (jan 2018), https://trustedcomputinggroup.org/wp-content/uploads/TCG_Guidance_for_Securing_NetEq_1_0r29.pdf

[4] Sailer, R., Zhang, X., Jaeger, T., Van Doorn, L.: Design and implementation of a TCG-based integrity measurement architecture. In: USENIX Security symposium. pp. 223{238 (2004)

[5] Liqun Chen, Rainer Landfermann, Hans Löhr, Markus Rohe, Ahmad-Reza Sadeghi, and Christian Stüble. 2006. A protocol for property-based attestation. In Proceedings of the first ACM workshop on Scalable trusted computing (STC '06). Association for Computing Machinery, New York, NY, USA, 7–16. DOI:https://doi.org/10.1145/1179474.1179479

[6] De Benedictis, M., Lioy, A.: Integrity verification of Docker containers for a lightweight cloud environment. Future Generation Computer Systems 97, 236{246 (2019)

[7] Luo, W., Shen, Q., Xia, Y., Wu, Z.: Container-IMA: A privacy-preserving Integrity Measurement Architecture for Containers. In: 22nd International Symposium on Research in Attacks, Intrusions and Defences (fRAIDg 2019). pp. 487{500 (2019

[8] Chen, L., Lohr, H., Manulis, M., Sadeghi, A.R.: Property-based attestation without a trusted third party. In: International Conference on Information Security. pp. 31{46. Springer (2008)

[9] Giannetsos, T., Krontiris, I.: Securing v2x communications for the future: Can PKI systems offer the answer? In: 14th Int. Conf. on Availability, Reliability and Security. ARES '19 (2019)

[10] The FutureTPM Consortium, “D4.1 – Threat Modelling & Risk Assessment Methodology,” 2019

[11] The FutureTPM Consortium, “D6.1 – Technical Integration Points and Testing Plan,” 2019

[12] The FutureTPM Consortium, “D2.1 – Second Report on New QR Cryptographic Primitives,” 2019.

[13] The FutureTPM Consortium, “D4.2 – FutureTPM Risk Assessment Framework – First Release,” 2019

[14] The FutureTPM Consortium, “D1.1 - FutureTPM Use Cases and System Requirements,” 2018

[15] The FutureTPM Consortium, “D5.1 – First Version of Implementation,” 2019

[16] The FutureTPM Consortium, “D6.3 – Demonstrators Implementation Report – First Release,” 2020

[17] The FutureTPM Consortium, “D4.4 – FutureTPM Risk Assessment Framework,” 2020

[18] The FutureTPM Consortium, “D4.5 – “Runtime Risk Assessment, Resilience and Mitigation Planning”, 2020

[19] The FutureTPM Consortium, “D6.5 – “Final Demonstrators Implementation Report”, 2020

[20] The FutureTPM Consortium, “D2.3 – “Final Report on New QR Cryptographic Primitives”, 2020

[21] The FutureTPM Consortium, “D6.4 – “FutureTPM Integrated Framework”, 2020

[22] The FutureTPM Consortium, “D4.3 – “Runtime Risk Assessment, Resilience and Mitigation Planning– First Release”, 2020

[23] The FutureTPM Consortium, “D5.1 – “First version of implementation”, 2020

[24] The FutureTPM Consortium, “D5.1 – “First version of implementation”, 2019

[25] The FutureTPM Consortium, “D5.2 – “Second version of implementation”, 2020

[26] The FutureTPM Consortium, “D5.5 – “Coprocessor Demonstrator”, 2020

[27] The FutureTPM Consortium, “D2.2 – “Second Report on New QR Cryptographic Primitives”, 2020

[28] The FutureTPM Consortium, “D3.1 – “First Report on Security Models for the TPM”, 2019

[29] The FutureTPM Consortium, “D3.2 – “First Report on the Security of the TPM”, 2020

[30] The FutureTPM Consortium, “D3.3 – “Second Report on Security Models for the TPM”, 2020

[31] The FutureTPM Consortium, “D3.4 – “Second Report on the Security of the TPM”, 2020

[32] The FutureTPM Consortium, “D3.5 – “Final Report on the Design and Security of the QR TPM”, 2020

[33] GALATICS Repository. Website: https://github.com/espitau/GALACTICS

[34] ProVerif: Cryptographic protocol verifier in the formal model - [online] https://prosecco.gforge.inria.fr/personal/bblanche/proverif/

[35] Tim Fritzmann, Georg Sigl, and Johanna Sepúlveda. RISQ-V: tightly coupled RISC-V accelerators for post-quantum cryptography. IACR Trans. Cryptogr. Hardw. Embed. Syst., 2020(4):239–280, 2020.

[36] Goldman, K. (2017). Attestation Protocols. Technical report, IBM, December 2017. https://developer.ibm.com/technologies/linux/articles/trusted-boot-openpower/

[37] Trusted Computing Group (TCG). TPM2.0 library specification – part 3 – commands – code. Available at: https://trustedcomputinggroup.org/wp-content/uploads/TCG_TPM2_r1p59_Part3_Commands_code_pub.pdf

[38] Dolev, D., Yao, A. (1983). On the security of public key protocols. IEEE Transactions of information theory, 29(2): 198-208.

[39] Basin, D., Cremers, C., Dreier, J., Meier, S., Sasse, R., Schmidt, B. (2019). Tamarin prover (v.1.4.1). https://tamarin-prover.github.io/

[40] Shao, J., Qin, Y., Feng, D. (2018). Formal analysis of HMAC authorization in the TPM2.0 specification. IET Information Security, 12(2):133-140.

[41] Shao, J., Qin, Y., Feng, D., Wang, W. (2015). Formal analysis of enhanced authorization in the TPM2.0. In proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, p.273-284.

[42] Lowe, G. (1997). A hierarchy of authentication specifications. In proceedings of 10th Computer Security Foundations Workshop, p.31-43. IEEE, 1997.

[43] Sassu R., Vlasceanu S., Simple Remote Attestation with Secure & Attested Communication Channels. Linux Security Summit Europe 2019. Available at: https://static.sched.com/hosted_files/lsseu2019/bd/secure_attested_communication_channels_lss_eu_2019.pdf