CONVOLVE

Seamless design of smart edge processors

GRANT AGREEMENT NUMBER: 101070374

Deliverable D3.1

Requirements, Threats, and Vulnerabilities Analysis

Title of the deliverable	Requirements, Threats, and Vulnerabilities Analysis
WP contributing to the deliverable	WP 3
Task contributing to the deliverable	Task 3.1
Dissemination level	PU – Public
Due submission date	30/04/2023
Actual submission date	28/04/2023
Author(s)	Alejandro Garza (NXP), Adrian Marotzke (NXP), Sven Argo (RUB), Jan Richter-Brockmann (RUB), Benjamin Cramer (BOS), Marc Geilen (TUE), Mottaqiallah Taouil (CIC), Lara Arche (TASE)
Internal reviewers	Manil Dev Gomony (TUE) Mounir Ghogho (UIR)

Document Version	Date	Change
V0.1	17/03/2023	Initial Document and Table of Contents
V0.2	23/03/2023	Document Adapted to Agreed Structure and Contributions
V1	19/04/2023	First Draft with contributions
Final	28/04/2023	Changes implemented from Review Process

Table of Contents

Deliverable Summary

This document gives a concise and comprehensive description and definition of adversarial capabilities and limitations, threats, potential vulnerabilities, and targets, derived from the applications and use-case scenarios, as defined in (WP1). It formulates attacker profiles that can be consulted during security evaluation and assessment, to confirm the provided security guarantees of the novel TEE architecture with its Post-Quantum Crypto (PQC) and CIM accelerators.

1. Objectives

This document "D3.1 Requirements, Threats, and Vulnerabilities Analysis" is a deliverable of the Working package No. 3 "Composable Real-Time and Hardware Security", task T3.1 "Requirements, Threats, and Vulnerabilities Analysis" under the task lead of NXP.

CONVOLVE aims to research and develop ultra-low-power secure processors for edge devices. Specifically, this deliverable is related to the objective of providing hardware security solutions against known attacks and to future proof the developments by incorporating Post Quantum Crypto security. Furthermore, the use of Trusted Execution Environments and Composable Security is also present in this deliverable.

This deliverable seeks to explore the vulnerabilities and threats associated with the edge computing solutions that will be developed in the project. It proposes solutions to mitigate these security risks and considers the security requirements derived from the use cases defined in CONVOLVE.

The findings derived from the analysis will serve as input for the Work in WP 3 to create the security solutions.

a) WP3 Objectives

- Detection and prevention of physical/hardware attacks, including side-channel analysis and fault injection attacks, through ultra-low-power protection mechanisms.
- Design of ultra-low-power, real-time, modular, and composable, long-term quantum-secure TEE (Trusted Execution Environment) for RISC-V processor architectures.
- Extend the TEE with secure hardware accelerators to achieve ULP long-term security using quantum-secure crypto cores and secure computation-in-memory (CIM) based neuromorphic computing.

b) Deliverable Structure

In this document, we will focus to adopt a bottom-up approach that focuses first on the technological security developments that will be explored in CONVOLVE. This is because the use cases that have been identified in WP 1.1 do not provide such strong security requirements that map successfully with what it is wanted to be researched in security. It should be noted that throughout this project we seek to develop solutions that are future-proof against known emerging security threats to edge devices.

However, the use cases will be considered to provide guidelines for the development of the different elements in WP 3. Similarly, the different technologies such as TEE, PQC, CIM, and Composability will have a positive effect and can be used in the context of the CONVOLVE use cases. With this, even if the use cases extend to similar applications in more demanding security contexts the proposed solutions will still address the arising needs.

To address the approach described above, the document will take the following structure.

- Identification of Emerging and Potential vulnerabilities and threats in the context of the Project and Use Cases
- Proposed Solutions
- Use Cases addressed in the security context
- Adversarial Models
- High Level Overview of the Work in WP 3
- Conclusion

2) Threats and Vulnerabilities

This section explores threats and vulnerabilities related to edge devices in the context of CONVOLVE. Since in this deliverable we are taking a bottom-up approach, the current threats, and vulnerabilities of the different security solutions to be developed in the project are also described. New security techniques are not immune to failures and problems, throughout the project we will seek to consider these to create more robust solutions.

In the Threats subsection, those related to Edge computing devices are presented. Similarly, the threat to cybersecurity represented by the emergence of a quantum computer is detailed. Additionally, it covers how Computation in Memory seeks to improve the efficiency of AI at the Edge, but nevertheless this computing paradigm can also be subject to attacks. Finally, Trusted Execution Environments are discussed, as they are one of the most used solutions for the security of embedded systems, but nevertheless are not immune to different attacks.

Finally, the Vulnerabilities are mentioned for the different computing paradigms and security solutions that will be used in CONVOLVE.

a) Threats

i) Edge Computing

Edge computing refers to enabling technologies to perform computations as close as possible to the data sources. The edge is defined as any network or computing resource between the data sources and the cloud. In general, as edge devices are deployed on the field, they must have higher portability and smaller size than personal computers and servers. These trade-offs limit the devices in their memory, energy, and computational capabilities [1]. Additionally, they have communication accessibility to connect to the internet and to other devices.

The increase of devices on edge has also generated an increase in real-world data, which can be combined with AI to create meaningful solutions. AI involves a huge amount of data to train the models, find patterns, improve, and customize the use of models. When sensitive information such as training data, inference results, or the parameters and hyperparameters of the model have been shared across different entities different privacy concerns arise[2].

The limitations in power, computational resources, and memory must not prevent the efficient implementation of the security requirements. Eventually, edge devices must provide throughout its lifecycle security, privacy, safety, reliability, and resiliency to become trustworthy and widely adopted. They can be susceptible to several types of attacks such as malware attacks, physical attacks, supply chain attacks or denial of service (DoS) attacks.

[1] Shi, Weisong, et al. "Edge computing: Vision and challenges." IEEE internet of things journal 3.5 (2016): 637-646.

[2] Mireshghallah, Fatemehsadat, et al. "Privacy in deep learning: A survey." arXiv preprint arXiv:2004.12254 (2020).

ii) Quantum Computer

With the rapid advances in physics, and in particular quantum physics, scientists have devised a fundamentally new model for computers. Opposed to the currently ubiquitous *classical computer* which works with binary states, i.e., power on vs. power off, *quantum computers* work with *qubits*. Qubits adhere to the complex laws of quantum mechanics and upon measurement, they collapse down to one of two distinguishable states. This allows the implementation of algorithms which work on multiple states simultaneously and consequently yields solutions to previously intractable problems.

Two fundamental problems of modern cryptography, namely the integer factorization problem and the discrete logarithm problem, are directly affected. More concretely, if a sufficiently large and powerful quantum computer is built, all cryptographic protocols based on either of these problems provides no security guarantees anymore.[3] At the time of this writing, only small, prototypical quantum computers have been constructed in isolated environments and are not available to adversaries. Nonetheless, it is inevitable to consider adversaries with access to large-scale quantum computers as a distinct threat today. The reason is that sensitive data produced now may still be equally sensitive in thirty years or more. Given that it is hard to estimate how long exactly it will take until quantum computers are available, we must ensure that all data is appropriately protected. This can be achieved with post quantum cryptography which is further detailed in Section 4.

iii) Physical Attacks

Physical attacks like power side-channel attacks[4] and fault-injection attacks[5] pose a huge threat against cryptographic algorithms implemented on embedded hardware devices including microcontroller, ASICs, and FPGAs. In case an attacker has physical access to a target device, she can acquire power traces during an encryption or decryption process by directly connecting an oscilloscope to the power supply of the chip or measuring the electromagnetic radiation[6]. Since the dynamic power consumption of modern CMOS technologies highly depends on the switching activities of the integrated transistors processing secret key material, an adversary can extract this information from the acquired power traces. This attack vector is considered as a passive attack.

In contrast, fault-injection attacks are counted to active attacks since an adversary needs to inject a fault into an ongoing encryption or decryption process altering an intermediate state of the underlying cryptographic algorithm. Utilizing the faulty (or even correct) outputs of the target algorithm, allows the adversary to learn information about the secret key material applying some statistical analysis.

[3] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," SIAM Rev., vol. 41

[4] P. C. Kocher et al., "Differential Power Analysis," in Advances in Cryptology - CRYPTO '99

[5] E. Biham et al., "Differential Fault Analysis of Secret Key Cryptosystems," in Advances in Cryptology - CRYPTO '97

[6] K. Gandolfi et al., "Electromagnetic analysis: Concrete results," in Cryptographic Hardware and Embedded Systems - CHES 2001

Over the last two decades, many different fault-injection mechanisms have been presented from researchers from academia and industry. These techniques range from simple clock or voltage glitches to more sophisticated methods using electromagnetic pulses to highly advanced approaches using lasers[7].

iv) Computation-in-Memory

Conventional processing units based on von Neumann architecture are not suitable for Artificial Intelligence (AI) or Machine Learning (ML) based applications where power efficiency, data latency and parallelism are key factors [8]. These factors become further relevant when the AI acceleration is to be performed by an Edge device. Special hardware accelerators are required to process ML algorithms that are energy efficient and execute in parallel. Computation-in-Memory (CIM) based processing units are best suited for data-intensive applications such as machine learning and data analytics. CIM architecture enables the instructions to be executed within the memory without the need to be sent for processing[9]. Non-volatile memory such as memristors based CIM architectures are best suited for Edge devices where pre-trained Neural network models are implemented within memory cells to perform inference[10]. But just like all other CMOS devices, these CIM devices are also susceptible to hardware security threats[11].

Among all other threats, side-channel attacks pose threats to confidentiality of the system in the form of Reverse Engineering and Data theft. Side-channel attacks are non-invasive attacks and attacker just have to be possessed the device or be in close vicinity to it which is quite possible in case of edge devices[12]. Vital information regarding architecture of the CIM implementation can be extracted through Power analysis. Timing and statistical analysis of the power traces have been used in literature to successfully reverse the neural network implementation on Microcontrollers, FPGAs and most recently CIM architectures[13]. Once the architecture is completely known, it is also possible to steal the input data to the system which may possess sensitive information like in the case of medical images[12].

[7] S. P. Skorobogatov and R. J. Anderson, "Optical Fault Induction Attacks," in CHES

[8] Y. Chen, Y. Xie, L. Song, F. Chen, and T. Tang, 'A Survey of Accelerator Architectures for Deep Neural Networks', Engineering, vol. 6, no. 3, pp. 264–274, 2020

[9] A. BanaGozar et al., 'CIM-SIM: Computation In Memory SIMulator', in Proceedings of the 22nd International Workshop on Software and Compilers for Embedded Systems, Sankt Goar, Germany, 2019

[10] W. Wan et al., 'A compute-in-memory chip based on resistive random-access memory', Nature, vol. 608, no. 7923, pp. 504–512, Aug. 2022

[11] S. Sayyah Ensan, K. Nagarajan, M. N. I. Khan and S. Ghosh, "SCARE: Side Channel Attack on In-Memory Computing for Reverse Engineering," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 29, no. 12, pp. 2040-2051, Dec. 2021

[12] M. Méndez Real and R. Salvador, "Physical Side-Channel Attacks on Embedded Neural Networks: A Survey," Applied Sciences, vol. 11, no. 15, p. 6790, Jul. 2021, doi: 10.3390/app11156790

[13] Z. Wang, F. Meng, Y. Park, J. Eshraghian and W. Lu, "Side-Channel Attack Analysis on In-Memory Computing Architectures" in IEEE Transactions on Emerging Topics in Computing, vol. , no. 01, pp. 1-13

v) TEE

Trusted Execution Environments (TEE) aim at providing additional security by isolating high-risk software from untrusted code. An example for such high-risk software could be the cryptographic components that ensure the confidentiality and integrity of an ML model. A typical operating system (OS) in this example would be untrusted, as OS are complex, often involving millions of lines of code which are almost certain to contain bugs. Thanks to the hardware enforced isolation, a TEE would ensure that even the OS could not access the secrets inside the TEE.[14] [15]

Although TEEs jointly use secure hardware and software mechanisms, they are not immune to attacks[16]. Two common attacks against TEE implementations are cache attacks and side-channel attacks. Cache attacks are a specialised form of side-channel attacks. These types of attacks target additional, non-intended inputs and outputs of the TEE, such as timing, power, electromagnetic-emanation, or voltage information.

Cache attacks target the timing variance in memory accesses, which depend on whether data is included in the memory cache of a processor or not. As the TEE also reads and write to the main memory via the cache, such timing variation can be used to extract data from the TEE by cleverly preparing the processors cache in a malicious way.

Another example of a side-channel attack is modifying the voltage of the processor while the TEE is running, thereby inducing faults that e.g., cause encryption operations to be skipped, thus exposing information. However, over the years, many more attacks against TEEs have been discovered. An overview of such attacks is displayed in Figure 1. Overview of TEE vulnerabilities [16] below.

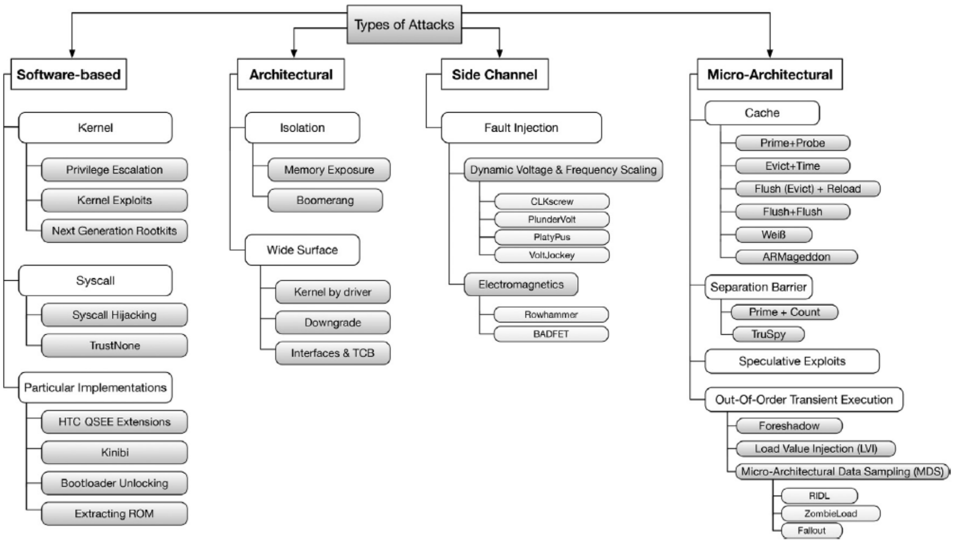


Figure 1. Overview of TEE vulnerabilities [16]

[14] D. Lee et al., "Keystone: An open framework for architecting trusted execution environments," in Proceedings of the Fifteenth European Conference on Computer Systems, 2020, pp. 1–16

[15] D. Kohlbrenner et al., "Building open trusted execution environments," IEEE Security & Privacy, vol. 18, no. 5, pp. 47–56, 2020

[16] A. Muñoz et al., "A survey on the (in)security of trusted execution environments". Computers & Security, Volume 129, 2023,

While TEEs are not immune to attacks, research and countermeasures have been refined over the years. However, most of the most used TEEs are closed source, which could slow down their development.

b) Vulnerabilities

i) Edge Computing

Below are some of the top examples of vulnerabilities and challenges that affect edge devices.

Authentication. - The data provided to the devices should come from a trusted source, this data can be software messages, or other information. Attackers can exploit ineffective authentication mechanisms.[17]

Insecure Ecosystem Interfaces. - The interfaces in the ecosystem outside the device can compromise it, and other associated elements. Examples of these interfaces can be APIs, cloud, web, and mobile that can themselves have security issues that affect the edge device.[18]

Insufficient Privacy and Data Protection. - Data collection and storage should follow a policy and practices to ensure confidentiality. Additionally, the use of cryptographic protection and suppression of data relations helps to guarantee anonymity [19]. This lack of encryption or access control mechanisms facilitates the access to data during transit, rest, and even processing by unauthorized users.

Lack of Device Lifecycle Management. - It is necessary to administer the devices deployed in the field. The device manager component should provide monitoring, update management, secure decommissioning, and support throughout the device lifecycle.[20]

Resource Exhaustion. - One of the main constraints of edge devices is limited energy, and for some of them, the incapacity to replenish it. Operations like the firmware update and encryption can cause the depletion of energy for the device, therefore these processes must be done with suitable mechanisms considering the trade-offs.

Secure Update Mechanism. - Many of the edge devices can easily become compromised and due to the lack of suitable update mechanisms, these may remain unpatched after a long time of being deployed in the field. It should be emphasized that this system must be well designed, as the updates can also become attack vectors.[21]

[17] Neshenko, Nataliia, et al. "Demystifying IoT security: an exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations." IEEE Communications Surveys & Tutorials 21.3 (2019): 2702-2733

[18] OWASP internet of Things project - OWASP.", [Online]. Available:https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Top_10

[19] Boulemtafes, Amine, Abdelouahid Derhab, and Yacine Challal. "A review of privacy-preserving techniques for deep learning." Neurocomputing 384 (2020): 21-45.

[20] Zhang, Zhi-Kai, et al. "IoT security: ongoing challenges and research opportunities." 2014 IEEE 7th international conference on service-oriented computing and applications. IEEE, 2014

[21] K. Zandberg, K. Schleiser, F. Acosta, H. Tschofenig, and E. Baccelli, "Secure firmware updates for constrained iot devices using open standards: A reality check," IEEE Access, vol. 7, pp. 71 907–71 920, 2019

ii) PQC Vulnerabilities

Post quantum cryptography is a flourishing and promising research area and an effective countermeasure against emerging quantum computers. However, it is also young and less well-studied. For instance, during the NIST standardization process more than fifteen schemes have been completely broken[22], significantly attacked, or withdrawn. This clearly indicates that PQC must be used with caution in practical applications to ensure that no vulnerabilities are inadvertently introduced[23].

Ideally, post quantum cryptography should be coupled with "classic" cryptography as explained in Section 3 to mitigate the sudden changes of security guarantees. Another critical aspect is the implementation of the scheme. As mentioned in Section 2, adversaries with physical access to the devices which performs cryptographic operation can use side channels like the power consumption to obtain information about secret values. Consequently, the schemes must be implemented in a side-channel resistant manner. This is, however, not always easily (or at all) and efficiently possible[24] and is tightly coupled to the operations and data types used internally.

iii) Computation-in-Memory

CIM architecture possesses the capability of true parallelism due to its tiled architecture and handling the vector matrix multiplication in a single clock cycle[25]. These features make it inherently safe from sequential timing analysis attacks. On the other hand, the replication of fixed tiles positions and use of power-hungry ADCs add some vulnerabilities to the design. Addition of ADCs into the crossbar array enables CIMs to produce vector matrix multiplications within a single clock cycle. ADCs in this case consume major portion of the energy required by CIM, which becomes an identification signature in the power trace to perform side-channel attacks.

Memristor based CIM architectures are best suited for implementation of Neural Networks due to their non-volatile capability to store weights in the resistance states of the memristors[26]. These weights and NN architecture are vulnerable to side-channel attacks. An adversary with the architectural knowledge of the CIM may use simple power analysis, statistical power analysis in combination of supervised or learning based attack to replicate the exact NN implementation within the CIM crossbars. Therefore, it is necessary to find all possible vulnerabilities and implement the energy and area efficient countermeasures.

[22] W. Castryck et al., "An efficient key recovery attack on SIDH", Advances in Cryptology-EUROCRYPT 2023

[23] https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Brochure/quantum-safe-cryptography.html?nn=916626

[24] S. Kundu et al., "Higher-order maskedsaber," in Security and Cryptography for Networks: 13th International Conference, SCN 2022

[25] A. Haron, J. Yu, R. Nane, M. Taouil, S. Hamdioui and K. Bertels, "Parallel matrix multiplication on memristor-based computation-in-memory architecture," 2016 International Conference on High Performance Computing & Simulation (HPCS), Innsbruck, Austria, 2016, pp. 759-766

[26] M. A. Zidan, J. P. Strachan, and W. D. Lu, "The future of electronics based on memristive systems," Nature electronics, vol. 1, no. 1, pp.22–29, 2018

iv) TEE

One of the challenges of commercial TEEs is that they provide almost no flexibility for different use cases or security demands. They are commonly designed for specific hardware and the architecture is driven by the security concerns of potential customers. The proprietary implementations are closed source, meaning there are few options to modify and tailor these solutions. It is challenging for hardware vendors to give all the details of a vulnerability, as this will most likely involve disclosing intellectual property of the hardware architecture [27].

A quick search of the Common Vulnerabilities and Exploits (CVE) database returns 72 results for ARM TrustZone and 35 for SGX. Which are the two prominent technologies for TEEs. Software-level vulnerabilities in most of the system components do not the TEE as in principle these components are considered untrusted. Nevertheless, a bug in trusted components and hardened code can compromise the TEE guarantees. Therefore, is necessary to have a secure update mechanism. Implementation Flaws occur when the TEE is not properly implemented, or its associated security mechanisms are correctly configured, exposing the TEE to attacks [28].

[27] Kohlbrenner, David, et al. "Building open trusted execution environments." IEEE Security & Privacy 18.5 (2020): 47-56

[28] Sabt, Mohamed, Mohammed Achemlal, and Abdelmadjid Bouabdallah. "Trusted execution environment: what it is, and what it is not." 2015 IEEE Trustcom/BigDataSE/Ispa. Vol. 1. IEEE, 2015

3) Roadblocks

Here are some of the roadblocks to generate and implement countermeasures against current and emerging threats for Edge devices in the context of CONVOLVE.

- New PQC schemes are in early phase of getting selected and standardize by NIST
- Hybrid schemes (PQC and traditional) introduce redundancy which directly opposes the limited resource constraints.
- Key size requirements for PQC can become a barrier for the implementation in resource constrained devices.
- Deployability on low-power and memory limited devices of this new schemes will be challenging
- RISC-V maturity in the security domain is still not comparable to other established architectures
- Most utilized TEEs are closed source and have limitations in flexibility for being implemented in diverse hardware and software platforms.

4) Proposed Solutions

To counter the threat and vulnerabilities, as well as the described roadblocks we propose:

a) Trusted Execution Environments

A TEE is a tamper resistant isolated processing environment where applications are securely executed, and data can be securely stored even in an untrusted platform. By isolating sensible data such as encryption keys and sensible code like cryptographic libraries the attack surfaces are reduced. This separation helps to reduce the amount of code that is needed to be trusted for a secure operation. The sharing of hardware and other resources between trusted and untrusted components is addressed by the TEE.. The separation of the kernel has as main purpose to ensure isolation and enable the coexistence of different systems on the same platform.

Multiple implementations of TEE already exist, such as Intel SGX[29] and ARM TrustZone[30]. However, many of these implementations have been attacked over the years, due to e.g., implementation bugs or side-channel attacks. Open-source frameworks such as the Keystone project aim at provided secure, reliable, and yet still flexible TEE implementations for platforms such as RISC-V edge processors.

TEEs in RISC-V offer several advantages for development of solutions in the context of CONVOLVE. This is because they provide greater transparency as they are Opensource, customizability for different use cases and requirements, compatibility for different cores and software, as well as a modular architecture[31]. These benefits fit within the scope of CONVOLVE for the design of TEEs that address the different security, hardware and power constrains.

The fundamental Security Properties are covered within a modern TEE

- Confidentiality. - An attacker cannot have access to the data while in runtime within the TEE. Furthermore, confidentiality of security code and runtime states stored in memory must be preserved.
- Integrity. - Unauthorized entities cannot add, remove, or alter code while executing in the TEE. Also, integrity applies to data, as the data within the TEE cannot be tampered with.
- Authenticity. - Additionally, a TEE helps to address authenticity when integrity cannot be provided. This happens when an asset can be changed by an attacker, but the system is able to detect the changed asset before it is used and prevents a security fault.

The TEE is mainly composed of the following elements to carry out the protection of these properties[32].

[29] F. McKeen et al., "Innovative instructions and software model for isolated execution." Hasp@ isca, vol. 10, no. 1, 2013

[30] ARM Ltd., "ARM security technology building a secure system using TrustZone technology," Whitepaper, 2016

[31] Kohlbrenner, David, et al. "Building open trusted execution environments." IEEE Security & Privacy 18.5 (2020): 47-56.

[32] https://globalplatform.org/specs-library/tee-protection-profile-v1-3/

i) Secure Boot

Secure Boot ensures that only signed and verified firmware images are booted on a device. To ensure this, a read only first-stage bootloader checks and verifies the cryptographic signature of the OEMs firmware image, and only executes the firmware image if the signature is correct and corresponds to a public key in the read-only key store. The key store is loaded with the OEMs public key during the device manufacturing, and OEM uses the corresponding private key to sign the firmware images.

ii) Chain of Trust

The Secure Boot process implements a chain of trust. Meaning that the process starts with a trusted entity and the rest of the components can be authenticated before being executed by using cryptographic schemes. The initial trusted entity is referred as the Root of Trust (RoT). One common location is the ROM on SoC, as modifying or replacing this component by reprogramming attacks is complicated. Therefore, the RoT must be tampered resistant.

iii) Attestation

The TEE can provide proof that the environment is safe and has not been tampered by delivering cryptographical proof to a third party. This process relies also in RoT and can also provide firmware measurements and runtime states of the device to avoid impersonation.

iv) Trusted I/O Paths

Protects the authenticity of the communication between peripherals and the TEE. This is not covered by Intel SGX. This can be significant as edge devices require various peripherals such as sensors for their operation.

b) Post Quantum Cryptography

Post-quantum cryptography (PQC) is the collective term for the development, analysis and evaluation of cryptographic schemes which remain secure even with the advent of quantum computers. Thus, it is a key component of CONVOLVE in order to effectively counter the threat of *quantum computing* (see Section 2).

Also, post-quantum cryptography is a strict requirement to achieve *long-term security*. Even though no (sufficiently powerful) quantum computers exist yet, data that is generated today may remain sensitive for many years and it is thus crucial to integrate and apply post-quantum cryptography in practice now. While PQC can serve as a basis to achieve many more advanced security properties, we leverage it to guarantee *confidentiality*, *authenticity*, as well as *integrity* of data.

i) Key Exchange

Confidentiality is achieved by encrypting the transferred data. In this case, a potential eavesdropper is able to record the entire communication but not able to make any sense of it. In practice, encryption can be symmetric or asymmetric. The former is more efficient and more suitable for larger amounts of data. However, it requires that both parties possess one or more identical secret keys. The latter is less efficient but allows ad hoc data transfers in a confidential manner since no shared secret keys need to have been established previously.

Usually, asymmetric encryption is only used to establish or exchange symmetric keys. Most asymmetric encryption schemes, e.g., RSA or ElGamal, provide confidentiality, but are not secure against quantum computers, since they are based on either the factorization or discrete-logarithm problem. Consequently, they fail at providing long-term security.

Post-quantum schemes, on the other hand, are secure against quantum computers but are immature and potentially brittle. Two prime examples are Rainbow and SIKE which have been broken completely even on non-quantum computers. Nonetheless, both schemes have passed multiple rounds of the NIST PQC Competition and their security flaws have passed numerous scrutinies by experts over several years. The resolution is a *hybrid key exchange*, which is a parallel combination of "classic" and PQC key exchange. Both shared "keys" are eventually fed into a key derivation function to obtain the actual symmetric key.[33] If an adversary can break one of the schemes, for example by building a quantum computer, the final symmetric key is still secure, and the confidentiality of the exchanged data is guaranteed.

ii) Payload Encryption

Once two parties possess one or more (identical) secret keys they can exchange data confidentially by encrypting it symmetrically. For many years, the Advanced Encryption Standard (AES) has been one of the most widely used symmetric encryption primitives. It is very mature, well-studied, and several highly optimized implementations exist. AES provides up to 256-bit of security against classical computers and is also secure against quantum computers for the foreseeable future, which makes it a valuable asset to achieve both confidentiality and long-term security.[34]

Additionally, there exist variants to obtain an *authenticated encryption* scheme. In this case, the data is not only kept confidential but also protected against random or malicious modifications. If only the two communicating parties have access to the secret keys, an authenticated variant of AES can thus be used to achieve confidentiality, integrity, and authenticity, which minimizes potential threats and attack vectors.

[33] A. Giron et al., "Post-quantum hybrid key exchange: a systematic mapping study," Journal of Cryptographic Engineering, vol 13

[34] X. Bonnetain et al., "Quantum Security Analysis of AES", IACR Transactions on Symmetric Cryppology, 2019

iii) Digital Signatures

Digital signatures provide a means of guaranteeing the authenticity of data. In other words, they allow the receiver to verify that the author (or sender) is who he/she claims to be. Digital signatures are an asymmetric cryptographic primitive and no secret values have to be established beforehand. This qualifies digital signatures to be used in ad hoc communications and a wide range of practical use cases, e.g., remote updates. They also emerge in the context of TEEs and secure boot (see Section 4). Similarly, to asymmetric encryption, classic signature schemes fail at providing the required long-term security, if quantum computers are built.

iv) PQC Schemes and Official Recommendations

Numerous PQC schemes for asymmetric encryption and digital signatures have been devised in academic and industrial research and the national standardization efforts such as the NIST PQC standardization process promoted this even further.[35] Most of the schemes have been extensively studied and analysed with regard to security, efficiency, simplicity as well as other criteria. However, the research and standardization procedures are still ongoing, and recommendations may change at short notice. Table 1 provides an overview of some more prominent post-quantum schemes. The second column indicates whether the scheme provides digital signatures (DS) or means of establishing a key (KEM/PKE) and the last columns indicate the current status with regard to standardisation and official recommendation of the scheme.

Scheme	Type	Standardization Process	Recommending Organisations
CRYSTALS-Kyber	KEM/PKE	NIST (selected)	NIST, ANSSI
CRYSTAL-Dilithium	DS	NIST (selected)	NIST, ANSSI
FALCON	DS	NIST (selected)	NIST, ANSSI
SPHINCS+	DS	NIST (selected)	NIST
BIKE	KEM/PKE	NIST (Round 4)	NIST$^{?}$
Classic McEliece	KEM/PKE	NIST (Round 4)	NIST$^{?}$, BSI
HQC	KEM/PKE	NIST (Round 4)	NIST$^{?}$
FrodoKEM	KEM/PKE	NIST (Round 3)*	BSI, ANSSI
NTRU Prime	KEM/PKE	NIST (Round 3)*	
HAETAE	DS	KPQC (Round 1)	KPQC$^{?}$
SMAUG	KEM/PKE	KPQC (Round 1)	KPQC$^{?}$
TIGER	KEM/PKE	KPQC (Round 1)	KPQC$^{?}$

Table 1: Selection of PQC schemes and their current status with regard to standardization and official recommendation. Entries marked with * indicate that the scheme is no longer considered for standardization by the institute and those marked with $^{?}$ are still under review.

For CONVOLVE, a combination of well-studied, secure, and energy-efficient schemes will be targeted. Standardized and officially recommended schemes are preferred to facilitate the adoption and interoperability. The precise choice as well as the practical realization are two of the central research points in WP3 for the upcoming months.

[35] Homepage of NIST Standardization Process: https://csrc.nist.gov/projects/post-quantum-cryptography

c) Secure CIM

CIM architecture is unique to typical edge devices like microcontrollers and FPGAs, hence the countermeasures proposed for such devices are not suitable for CIM architectures[36]. In this context we propose to implement Artificial Neural Network in CIM device and propose effective countermeasures specific to CIM architecture. The vector matrix multiplications in CIM are performed within a single clock cycle using ADCs. These ADCs are the prime focus of adversaries to extract the multiplications results to reverse engineer the weights[37]. The goal is to eliminate leakage of sensitive information through ADCs. This can be achieved by adding a deterministic noise to the ADCs to make them less vulnerable. It is to note that only a single countermeasure is not enough to fool-proof the system from side-channel attacks.

Masking the power consumption among various sections of CIM architecture is also an effective countermeasure where the visual separation of various tasks performed is masked and logical separation of tasks cannot be determined. This can be achieved by adding dummy weights within the crossbar which are then eliminated at the later stage. Encrypting the incoming data and/or weights and decrypting them before generating output is also an effective way to secure the system from reverse engineering attacks using side-channel analysis.

d) Protection against Physical Attacks

Over the last two decades, many different countermeasures against power side-channel attacks and fault-injection attacks have been introduced. For CONVOLVE, protection against power side-channel attacks is especially interesting and should be the focus of this paragraph. In general, countermeasures against side-channel attacks can be divided into hiding- and masking-based approaches. While hiding tries to increase the noise or to decrease the signal, masking relies on provable secure techniques[38]. To this end, countermeasures based on masking techniques are preferred in CONVOLVE.

However, masking relies on secret sharing, i.e., a secret value x is split into s share such that$x = x_0 \circ x_1 \circ \ldots \circ x_{s-1}$. In case for group operator $\circ$ is realized as an exclusive or, the masking approach is called *Boolean sharing*. The group operator can also be replaced by additions or multiplications. In this case the masking approach is called *arithmetic sharing*. However, the desired security is achieved by generating $s - 1$ shares uniform at random and determining the remaining share such that the previous equation holds. Since all secret values are shared, the underlying functions need to be shared as well in order to process them secretly. Sharing linear functions can be accomplished in a straightforward manner by processing each share independently. However, adapting non-linear functions poses a challenge often connected with introducing overhead in terms of area, latency, and randomness requirements.

[36] S. Ghosh, M. N. I. Khan, A. De and J. -W. Jang, "Security and privacy threats to on-chip Non-Volatile Memories and countermeasures," 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, TX, USA, 2016, pp. 1-6

[37] Z. Wang, F. Meng, Y. Park, J. Eshraghian and W. Lu, "Side-Channel Attack Analysis on In-Memory Computing Architectures" in IEEE Transactions on Emerging Topics in Computing, vol. , no. 01, pp. 1-13

[38] Chari, S., Jutla, C.S., Rao, J.R., Rohatgi, P.: Towards Sound Approaches to Counteract Power-Analysis Attacks In: Advances in Cryptology - CRYPTO '99

The concept of masking as a countermeasure is well studied for symmetric ciphers like the Advanced Encryption Standard (AES). However, masking asymmetric ciphers and in particular PQC schemes is still a challenging task. Most PQC approaches use advanced mathematical structures and operations that require to research for efficient masking approaches. Depending on the application, the target countermeasure could be optimized with respect to latency, area (for hardware accelerators), and additional required fresh randomness. Especially for CONVOLVE, the power consumption plays a crucial role since the edge processor should be realized as an ultra-low-power device.

e) Composability

Research and design composable real-time platforms, compositional components for safety and security and associated modelling and synthesis methods.
CONVOLVE delivers a design flow that targets diverse accelerators for deep learning applications. Such diversity may also be reflected in the presence or absence of specific safety and security features. We therefore target a flexible set of components that can bring such safety and security features at affordable cost of power, area and design and verification requirement.
Such a solution may be provided with a set of compositional building blocks that can be freely put together, in accordance with the needs of a specific platform instantiation and assuring the services and requirements without additional costly verification.

This would specifically alleviate the need to review a particular constellation of safety and security components holistically, or even in combination with the full platform, its applications and components that rely on the TEE for their security.

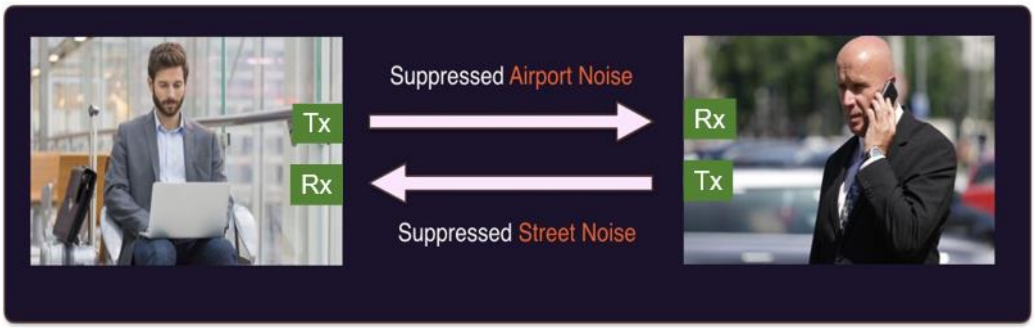
ULP platform cloud could potentially benefit from sharing of valuable resources, or from tight integration and co-optimization techniques. Such sharing of resources and tight integration potentially endangers safety, security, timeliness guarantees and may introduce risk of information leaking to other users of such shared resources on the platform or outside of the platform.

Challenges, ULP and resource efficiency may be at odds with composability and reusability. Optimization opportunities may be use-case specific and may have unintended consequences for the integrated system lowering security and potentially providing attack vectors.

We aim to exploit architectural HW/SW composability to develop novel solutions to deliver safety and security solutions at affordable cost in terms of power, resources, and design and verification time and effort. WP3 also intends to develop, as needed, modelling solutions to support and automate the design and synthesis flow that provides adequate solutions for CONVOLVE platforms without the need for costly design iterations and solution specific verification efforts. Such modelling solutions may be used for design-time automation, but also for run-time management of safety, security, or fault-tolerance features.

5) Use Case Requirements addressed by the WP

a) Deep Noise Suppression / Speech Enhancement



Deep *Noise Suppression (DNS)* or *Speech Enhancement* aims to improve the quality of both Tx (uplink) and Rx (downlink) speech signals by reducing background noise, thereby improving their quality or intelligibility.

i) Security Requirements and Considerations

In general, GNA is not overly concerned about overall lack of privacy or safety from the user's perspective, since no data is stored on the device, and data transmission commonly occurs within inherently insecure channels (air medium) or channels where security is ensured by the underlying transmission protocol (i.e., Bluetooth or other RF).
Mainly there are two considerations for security requirements:

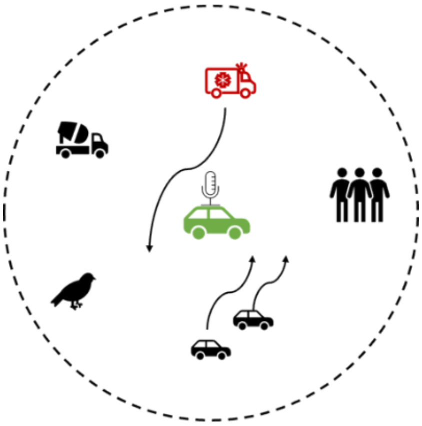
1. Protection of intellectual property in form of neural network models. Refers to the protection from copying or inferring of the specific neural network architecture and specialized data which can be expensive to acquire and train.
2. Secure update of firmware to the edge device. The process of updating the software that controls the hardware components of an (edge) device, such as a smartphone, IoT device or headset in a secure and trusted manner. This is important to ensure that the device remains secure and up to date, as vulnerabilities or bugs in the firmware could potentially be exploited by attackers to gain unauthorized access or cause damage to the (edge) device.
3. Secure update of AI Models. By using encryption and secure mechanisms to protect the transmission and storage of updates to neural networks, which are commonly used in machine learning applications. As before, this is important to prevent attackers from intercepting or tampering with the updates, which could lead to degraded performance, security vulnerabilities or IP infringements.

Additional requirement is that the in and out latency is done under 2ms.

ii) Adversarial Model for Use Case

We assume that an attacker will be able to have physical access to this type of edge device. They may be able to access, modify or extract the AI model from the device's memory or storage if it is not protected, in order to steal the underlying IP.

b) Acoustic Scene Analysis



This use case aims to predict the presence as well as the spatial position of emergency vehicles based on acoustic features recorded by microphones within typical traffic scenes.

i) Security Requirements and Considerations

Having detailed knowledge of emergency vehicles in the vicinity of a car could provide essential and safety-critical information, especially for autonomous driving. Hence, security must be deeply anchored in the design of any proposed solution for siren detection and tracking. Currently, there are three targets for a security consideration:

1. First, the **communication of the signals** from the microphone as well as the communication of the prediction of the model to other actors in the car needs to be secured in terms of attacks and privacy.
2. Second, the model itself might require **updates after the initial deployment** phase to respond to other input sounds – like sirens of different countries – or to output more detailed responses in potential refinements. This update process is a critical step impacting all future behaviour and could potentially be done remotely.
3. Last, the **model, i.e., the parameters need to be secured,** since they contain substantial knowledge and development effort.

Determined by typical time scales in traffic scenes, the use-case comes with throughput constraints. In more detail, the solutions should be able to perform a prediction at least every 100 ms. It is noteworthy, that the feature extraction already takes up 50 ms in the current implementation (without neural network processing) and hence a significant amount of the throughput limit. Further, a total power budget of les than 100 mW is available. Any proposed solution must adhere to these constraints that impose an upper limit for the whole implementation.

ii) Expand The Use Case

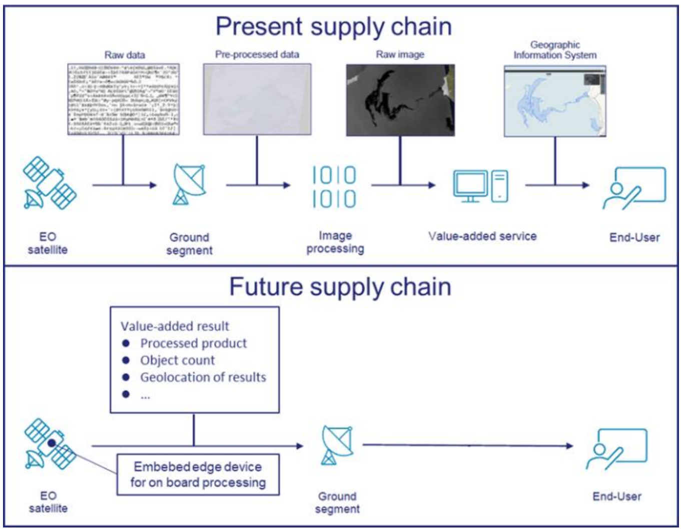
On a short timescale, the model could be continuously refined in terms of accuracy promoted by new training data. Beyond that, the capability of the neural network could be increased. This targets siren sounds of emergency vehicles of other countries, but, moreover, could also encompass the extension of new classes like regular cars.

Security within the aforementioned processes becomes even more significant in scenarios in which the prediction is not only used to inform the driver about the presence and location of an emergency vehicles, but an autonomous action needs to be taken based on the current prediction. This, however, is an essential step for autonomous driving and should hence be considered in the present use case and any associated security consideration.

iii) Adversarial Model for Use Case

We assume that an attacker will be able to have physical access to this type of edge device. Starting from the sensor, a possible attacker could target the blocking of the microphones. This in turn would lead to a dysfunctional system that is unable to signal any response. The same holds true for any removal of hardware components. An attacker could also try and extract the AI model, in order to steal the IP. Even more severe attacks target the manipulation of either input signals or model responses. Both strategies have the potential to lead to wrong actions with potentially detrimental outcome.

c) On Board Computer Vision



The images generated by Earth Observation (EO) satellites are traditionally downlinked to ground for processing and analysis. This causes congestion of the communications channel and the network itself due to the size of the information, which will not always be useful (as captured images sometimes don't include relevant information, for example scenes covered with clouds…), and represents a security breach for certain images. The use case pursues to implement a Supply chain that generates results added value results in the satellite reducing the number of steps.

i) Security Requirements and Considerations

Among other benefits, the change to an edge architecture for image processing on board the satellite will improve the security of the system not only in terms of technical implementation, but also in aspects related to information sovereignty.

The current operation scheme causes that certain confidential and/or sensitive images could be collected by third parties and even manipulated before reaching the analysis centre, causing an inappropriate decision making. However, by adding flexibility on board to receive the upload of new applications, a security risk is also introduced in the system, as it opens a door for hackers to introduce malware in the satellite.

To improve this scheme in terms of security it is necessary to ensure:

1. **The origin and integrity of the SW** being deployed on-board shall be always controlled and preserved, if needed including dedicated mechanisms to cover this.
   Different types of SW shall be taken into account:
   - Firmware
   - Applications
   - Neural Networks (AI models)

   In particular, it is essential to ensure the integrity of the infrastructure and platforms layers of the SW installed on board and to prevent potential malware from being uploaded from the ground. NNs deployed on board will undergo modifications during the lifetime of the device, then is necessary to ensure that updates from ground are performed securely and with integrity, allowing that only SW from a trusted origin is uploaded and deployed.

2. **Potential security vulnerabilities when downloading raw images** that could be compromised or modified (defence, cadastres, asset tracking,...). If only the result (the inference) is downloaded, it is more difficult to understand the value of the processed data for non-trained eyes, although it will be necessary to encrypt the information to prevent an attacker from using it. In the scenarios in which the enhanced image (value-added product) is downloaded, the security gap needs to be improved.

3. **Security of the communication channel.** The channel used to upload new SW or applications on the edge module and to transmit the information between the satellite and the ground is an over-the-air radio frequency link that needs to be secured against possible attackers who may use radio techniques to interfere with the uplink and/or downlink and modify the transmitted data. Part of this securitization takes place on the ground, but another part, the part responsible for the handling of the data by the satellite, and more specifically, the edge device, is relevant to this project.

There must be a power budget of less than 20 W per SoC considering all the computations and operations.

ii) Evolution of the Use Case

The use case will evolve throughout the life of the device, and the NN can even be changed so that the edge device performs very different processing activities on the images acquired during the project. For example, you might initially want to detect the number of ships in an image and later monitor certain regions for fires.

Additionally, the sharing of the device by different application users is contemplated in the future, so securing user access to the applications and data of others is a necessary requirement in the near future.

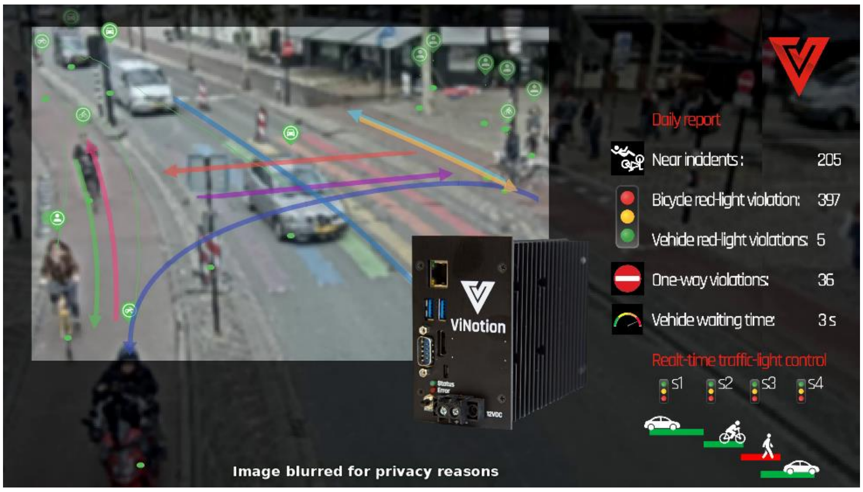
Another alternative, since this use case is the one that will consume the most resources, could go through a global architecture with several edge devices in parallel. In this case, it is necessary that the information between them must be shared securely. If all of them work in the same application and that the same protection schemes be considered as for the previous paragraph if they work grouped for different applications.

iii) Adversarial Model for Use Case

The systems must be secured and protected from firmware vulnerabilities to prevent attackers from gaining access to the system and compromising its security. The authenticity and integrity must be verified for every software or data received and sent by the satellite. However, it is assumed that the attacker does not have physical access to the satellite, as it is in orbit.

An example would be image spoofing attacks: An adversary might attempt to spoof satellite images by manipulating the images to hide or obscure objects. This can be particularly relevant when raw data is in transit from the satellite to the ground station.

d) Video-based traffic analysis


Daily report
Near incidents: 205
Bicycle red-light violation: 397
Vehicle red-light violations: 5
One-way violations: 36
Vehicle waiting time: 3 s
Realt-time traffic-light control
s1
s2
s3
s4
ViNotion
Image blurred for privacy reasons

ViSense is edge-based video analysis system that reads, analyzes and processes real-time video for surveillance, traffic management, incident detection, crowd management and various other traffic cases. By utilizing artificial intelligence (AI) software, ViSense makes it possible to register movements of all objects including pedestrians, bicycles, and vehicles.

i) Security Requirements and Considerations

1. **User Authenticated Access** with brute force protection and we support 3 levels of authorization (root, admin, viewer).
2. **Minimize sensitive and privacy data leaks** according to GDPR. Anonymization of images asap. That means we only show blurred images in the GUI and don't record video.
3. **Memory Encryption** to secure AI models, although they are decrypted while being in use.
4. **The firmware license** is connected to a **hardware fingerprint** to prevent illegal copying
5. **Secure publication of inference informatio**n. Currently done through and embedded VPN client.

A power budget of 20 Watts and must ensure real-time inference. If real-time is considered 20 frames per second, the processing should be done under 50 ms.

ii) Adversarial Model for Use Case

An adversary can have physical access to the device as they are placed on the street. In case the memory is not protected the attacker can have access to assets such as the AI model and the pre and post processing pipelines for the object detection model. An attacker could also try and modify the model, in order to falsify its outputs.

Attackers can compromise video devices remotely, allowing them to watch live camera streams, as well as compromise credentials to pave the way for future attacks. This must be avoided as these cameras can be placed in important infrastructure such as main ways on the cities and highways.

e) Powerful Adversary (Future Adversarial Model)

A powerful adversary for edge devices is an attacker who has significant resources and skills to compromise the security of the edge computing system. Such adversaries can be nation-states, criminal organizations, or highly skilled individuals who can launch sophisticated attacks. They might have access to advanced technologies in a future such as quantum computers or advanced machine learning algorithms, making their attacks more sophisticated and difficult to detect.

Additionally, they may have extensive knowledge of the edge device's hardware and software architecture, enabling them to identify and exploit vulnerabilities in the device’s firmware and software. Their goal could be to steal sensitive data (AI models, or privacy information about the users), sabotage operations, or gain unauthorized access to the device or network.

f) Security Requirements Mapping

The following Table 2 provides the main identified requirements that will be covered by the technology solutions explored in WP during CONVOLVE. These requirements are mapped to the previously described use cases.

Security Requirement	Rational	Use Case GNA	Use Case BOS	Use Case TASE	Use Case VIN
Secure Boot	Preventing unauthorized firmware from booting	X	X	X	X
Memory Encryption	Prevents other applications, external attackers from being able to access read memory	X	X		X
Long Term PQC Security	Devices will have long service life; QC may become a threat in that time		X	X	X
Hybrid Classic & PQC Security	New PQC are very new and untested, many were broken (see e.g. SIKE)	X	X	X	X
Timing Side channel resistance	Timing attacks such as cache attacks can remotely extract keys	X	X	X	X
Power side channel resistance	local attackers can use power analysis to also extract keys		X		
Fault Injection resistance	Deliberately induced faults can break crypto implementations				X
Encrypted & authenticated Communication	The communication between two devices (e.g. Two Bluetooth in-ears, or a satellite and ground station) is protected from eavesdropping and manipulation		X	X	X
Low power cryptography	Low power implementations lead to a longer battery life. Power hungry implementations may be disabled to save power, compromising security	X	X		
Composable Security Framework	Use-case owners can choose and select relevant features	X	X	X	X

TABLE 1

6) Overview of WP3 Work Plan

In this section we will broadly describe the work that is to be done throughout the project in the different tasks of WP 3. At the end, Table 2 shows the expected timelines and the most important milestones.

a) Task 3.2

Based on the outcomes of this document, we will work on the concrete evaluations, explorations, and specifications of security features. More precisely, our specifications consider the requirements of the use cases by focusing on security guarantees. Important parts throughout all specifications for all use cases are ultra-low power and real-time security features.

The preliminary specifications will contain a set of security features and countermeasures that should be implemented in a TEE providing long-term security. We plan to achieve this goal by aiming for a security level of AES-256, especially for symmetric schemes. Additionally, we plan to support hybrid schemes, i.e., a combination of classical asymmetric schemes and PQC schemes. Besides, we also investigate stand-alone PQC schemes inherently providing long-term security.

All these parts will be consolidated in a composable and extendable security framework. More precisely, this framework should be used to select required security features tailored to specific use cases with different security requirements. For example, in a scenario where CONVOLVE's edge processor is used in an environment where an adversary does not have physical access to the device, the processor does not need to be protected against power side-channel attacks.

b) Task 3.3

Task 3.3 addresses the implementations of the security features specified in Task 3.2. This includes hardware accelerators of symmetric, classical asymmetric, and post-quantum cryptographic algorithms. Furthermore, specific countermeasures against physical attacks are implemented as well adjusted to the use case requirements. Moreover, Task 3.3 addresses implementations of cryptographic algorithms for use cases with respect to CIM. All implementations -- classical and CIM -- will be designed to consume ultra-low power and to achieve real-time requirements. Eventually, all implementations will be incorporated into the composable security framework, implemented in a consolidated TEE, and attached to the RISC-V main processor.

c) Task 3.4

The security of the proposed TEE architecture and accompanying accelerators will be evaluated in this task against its requirements and specification. First, each component of the system will be evaluated in isolation (based on the best metrics for them). Finally, the complete system will be evaluated where possible.

This task deals with the security verification of the developed RISC-V based TEE architecture with its accelerators, as defined in tasks T2.2 and T2.3 and verify its security measures against the requirements and specification derived in WP1 and T2.1. The verification methodologies consist of tests that evaluate the requirements and KPIs.

Working Package 3	Participants	3	6	9	12	15	18	21	24	27	30	33	36
Composable real-time and hardware security		▓	▓	▓	▓	▓	▓	▓	▓	▓	▓	▓	▓
T3.1 Requirements, Threats, and Vulnerabilities Analysis	NXP, TUE, TASE, RUB, CIC, BOS, GNA, VIN	▒	▒										
T3.2 Security Architecture for TEE	RUB, TUE, NXP, UED, CIC			▒	▒	▒	▒	▒	▒	▒	▒		
T3.3 Compsable Implementation	RUB, TUE, NXP, UED, CIC					▒	▒	▒	▒	▒	▒	▒	
T3.4 Security Evaluation and Assessment	CIC, TUE, TASE, NXP, RUB					▒	▒	▒	▒	▒	▒	▒	
D3.1 Requirements, Threats, and Vulnerabilities Analysis	Lead NXP		■										
D3.2 Security Architecture and TEE Implementation	Lead TUE						■						
D3.3 Update of Security Architecture and TEE	Lead RUB											■	
D3.4 Component-Based Security Evaluation and Assessment	Lead CIC											■	

TABLE 2

7) Conclusion

The proposed security solutions/developments go beyond the requirements of the use cases. Nevertheless, it is mandatory to explore the use of this advanced techniques to develop future competencies that can accommodate in the spectrum of future uses of Edge Computing in diverse industries.

This analysis highlights the threats and vulnerabilities for edge devices. As well as the challenges for security solutions proposed in this project. Additionally, it considers the security requirements derived from CONVOLVE's use cases and their adversary models.

The findings derived from the analysis will serve as Input for the Work in WP 3. This deliverable will be updated in D3.3.