**Abstract**

**Module A.2 - FPGA based Visual Capture Device**

DeepShield: A distributed and cryptographic approach for authenticating digital content, based on Trusted Execution environments (TEE), establishing ownership, and detecting unwanted manipulations with focus on the Field Programmable Gate Array (FPGA) based visual capture device with integration with the overall DeepShield approach.

Stage 1 (Months 1–7, timeboxed, final demonstration 27/28th Mail 2025):

Research and develop core technologies; conduct initial testing and demonstration.

|  |  |
| --- | --- |
| Document No.: | **A2\_FPGA\_VCD\_Abstract** |
| Issue: | A |
| Date: | 05.05.2025 |

|  |  |
| --- | --- |
|  |  |

**Revision Record**

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Date** | **Description / Change** | **Modified Chapters** |
| A | 05.05.2025 | Delivered version | All |

Contact: langewisch@secublox.com

# **Table of contents**

[Table of contents 2](#_Toc198025006)

[Abstract 3](#_Toc198025007)

[Overall Goal DeepShield Approach 3](#_Toc198025008)

[Focus Module A.2 3](#_Toc198025009)

[List of Abbreviations 4](#_Toc198025010)

[DeepShield Overall Approach 5](#_Toc198025011)

[Implementation A.2 – FPGA VCD 6](#_Toc198025012)

[Requirements: Module A.2 – FPGA based Visual Capture Device 7](#_Toc198025013)

[Technical Concept: Module A.2 – FPGA based Visual Capture Device 8](#_Toc198025014)

[Scope of Work: Module A.2 – FPGA based Visual Capture Device 8](#_Toc198025015)

[Integration FPGA-Based Visual Capture Device with DLT 10](#_Toc198025016)

[Results A.2 – FPGA based Visual Capture Device 10](#_Toc198025017)

[FPGA based Visual Capture Device 10](#_Toc198025018)

[FPGA based Visual Capture Device Application 11](#_Toc198025019)

[DeepShield Integration Library 11](#_Toc198025020)

[Evaluation A.2 – FPGA based Visual Capture Device 12](#_Toc198025021)

[Demonstration A.2 – FPGA based Visual Capture Device 13](#_Toc198025022)

# Abstract

The rapid development of digital technologies in recent years has opened both fascinating and unsettling possibilities, particularly **in** synthetic content and deepfakes. This sophisticated AI-generated audio, image and video content is now so realistic that it is almost indistinguishable from real footage.

With the exponential increase in the performance of systems and language models based on artificial intelligence (AI) and machine learning (ML), not only the quantity but the quality of deepfakes is improving rapidly. This poses dangers – for trust, security and the perception of reality in our society.

Efforts to combat deepfakes are currently focused on two major areas:

* **Detection:** Development of AI algorithms to identify deepfakes
* **Prevention:** Implementation of authentication mechanisms for digital content.

Despite significant progress in these areas, major challenges remain, such as the generalizability and scalability of AI deepfake detection systems and the establishment of a manipulation-resistant standard for image metadata.

## Overall Goal DeepShield Approach

Development of a prototype that reliably detects deepfakes images from various media content and/or protects existing infrastructures from the use of deepfakes with preventive measures. To address these goals the overall DeepShield approach was divided between different independent modules (A.1, A.2, B, C, D) and will be integrated to complete the overall DeepShield approach, demonstrated at the end of stage 1.

## Focus Module A.2

This document focusses on the abstract for the design, concept, implementation and demonstration of module A.2 a prototype demonstration of the FPGA based visual capture devices and integration into the overall DeepShield approach.

**Technology:**

Prototype of the FPGA visual capture device will demonstrate and met following criteria to support the overall DeepShield goals:

* How image content can be reliably created and authenticated in real-time.
* Integration into the overall DeepShield approach
* Resilience in your team and in your solutions.
* Different use cases will show how authenticated and secured image data is registered in decentralized Blockchain network by which it is possible to use such content through different platforms like social media or news portals.
* Scalability and adaptability to different digital platforms

## List of Abbreviations

Below in Table 1 is a list of abbreviations used in this document.

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| BC | Blockchain |
| DLT | Distributed Ledger Technology |
| DN | Decentralized Network |
| DSC | DeepShield |
| IPFS | InterPlanetary File System |
| KYC | Know Your Customer |
| LLM | Large Language Model |
| NFT | Non-Fungible Token |
| PRNG | Pseudo-Random Number |
| TEE | Trusted Execution Environments |
| TRL | Technology Readiness Level |
|  |  |
|  |  |
|  |  |

## DeepShield Overall Approach

The DeepShield approach will focus on following requirements to demonstrate the defined goals by following functionalities:

|  |  |
| --- | --- |
| **A: Content Creation & Secure Watermarking** |  |
| A user captures a photo or video with on a public blockchain registered device.  These photos or videos are watermarked on-device within a secure, trusted execution environment making use of the private device key and then stored locally or elsewhere. |  |
|  |  |
| **B: Content Registration**  Additionally, to the preregistered public key of the devices, the original watermarked content can be registered on a public blockchain. This allows for ensuring context integrity using further meta information such as reference vector embeddings. |  |
| **C: Content Verification & Context Integrity**  The registered public key enables verification of the content’s authenticity, confirming its origin from a specific device based on the watermark. Degradations in the watermark indicate content manipulation.  Using meta information such as reference vector embeddings, contextual integrity can be verified, in a multilayered approach, helping to prevent disinformation. |  |
| **D: Verification and Revocation via Distributed Ledger**  Public Key Management: Verifiers can retrieve the device’s public key from the blockchain to authenticate the watermark and digital signature, ensuring transparent, verifiable content authenticity.  Revocation Notices: The ledger will contain information about revoked or compromised keys, ensuring that only active and trusted devices can be verified  Smart Contracts for Automation: Implement smart contracts to handle device registration, key updates, and revocation seamlessly |  |

# Implementation A.2 – FPGA VCD

To investigate the described challenges the DeepShield demonstrator for Stage 1 will be based on following overall approach divided into following modules:

A diagram of a blockchain network

AI-generated content may be incorrect.

Figure 1: DeepShield – Stage 1 – Main Modules

The overall DeepShield approach consists of the following main modules:

* **Module A (Create & Register Content)**: Authentication by integrating cryptographic, digital identity, watermarking together with FPGA and trusted execution environments logic (A.1: Smartphone app / A.2: FPGA-based machine). In additional signed data will be transferred to the decentralized network (Module B).
* **Module B (DLT Network):** Prototype of a decentralized network to register digital identities (devices, accounts), image signatures/integrity and image reference content. In additional to verify existing images.

**Module C (Verify Content):** Prototype of the Artificial Intelligence (AI) services to detec deep fakes of images connected with decentralized network.

* **Module D (Management & Integration):** Develop a prototype of the Integration-Framework (Dashboard, APIs, Integration)

This document focuses on Module A.2 – FPGA based Visual Capture Device.

## Requirements: Module A.2 – FPGA based Visual Capture Device

Module A.2 focuses on image content creation. For this a user captures an image with a FPGA based Visual Capture Devices, which authenticated and encrypt the image in real-time an registered such authenticated content on a public blockchain for further image processing by external platforms (e.g. social media or news portals).

A diagram of a brand identity

Description automatically generated

Figure 2: DeepShield – Module A.2 – FPGA Module

**Delivery focus for Stage 1 of Module A.2:**

The delivery of design, concept, implementation, validation and demonstration focuses on following aspects for module A.2:

* Use registered wallet (public / private key) for digital Identities (Owner, Device)
* Import private RSA keys for encryption of image content by FPGA image capture device
* Create digital image (together with metadata) by FPGA device
* Create digital watermark linked to created image
* Use DeepShieldLibrary to integrate with DeepShield approach for further image processing
* Create electronics: FPGA Modul connected to a Laptop by Ethernet-LAN

## Technical Concept: Module A.2 – FPGA based Visual Capture Device

This document outlines the technical specifications for the development of an FPGA-based visual capture device featuring real-time AES256 encryption for secure image processing and metadata handling.

A diagram of a computer

AI-generated content may be incorrect.

Figure 2: DeepShield – Module A.2 – Technical Concept Overview

The implementation and demonstration for Stage 1 focused on foundational capabilities. For this the architecture ensures image creation together with aligned metadata, robust encryption, efficient routing, and seamless blockchain integration while maintaining high performance and security standards.

## Scope of Work: Module A.2 – FPGA based Visual Capture Device

The scope of work to implement module A.2 is divided into following activities:

1. Image Capture Platform: Develop an FPGA platform capable of capturing high-resolution images.
2. A table containing keys for AES256 encryption is hardcoded into the device during Stage 1. This table ensuring that key access is strictly managed. The table enables efficient cryptographic operations by indexing keys via a unique Table ID, which is referenced in metadata for traceability and operational consistency. Furthermore, the table is synchronized with the routing (blockchain) node for decoding and tokenization of the metadata stream during data processing.
3. Real-Time AES256 Encryption: Implement hardware-accelerated AES256 encryption for:
   1. Container 1: Encapsulates image data, including an embedded predefined watermark in metadata.
   2. Container 2: Encapsulates metadata stream (CRC code, timestamp, device ID, and reference to key table via Table ID). The Table ID is the same for Container 1 and Container 2.
4. Key Management: Preload a key table for secure signing operations.
   1. Key indices are stored in metadata containers, with references to the table via a unique Table ID.
5. Data Routing: Design a two-tier system:
   1. Edge Device: Encodes both containers securely.
   2. Routing Node (Laptop): Processes data offloaded from the edge device by the DeepShield library to communicate with the blockchain network. This node will be authorized to the blockchain through a unique, assigned Node ID, ensuring secure and authenticated interactions with the blockchain.
      1. Decodes the metadata container (Container 2) and writes metadata (including Table ID and key index) to the blockchain network using the DeepShield library.
      2. Routes the image container (Container 1) to an IPFS storage system.
6. Integration Module B (DLT Network):
   1. Store public keys and metadata on the Blockchain network.
   2. Transition tokenized metadata on the Blockchain network.
7. Watermark Embedding: Configured by FPGA Image Capture Device Software to configure watermarking while image creation and additional setting.
8. Demonstration and Validation:
   1. Validate image capturing and AES256 encoding capabilities.
   2. Demonstrate metadata and image container offloading, and integration with Module B (DLT network) for further image processing along different platforms.
   3. Showcase the retrieval of secure image data and watermark and verification together with Module D: DeepShield-Management/Integration.

## Integration FPGA-Based Visual Capture Device with DLT

To connect with Blockchain network and register user accounts, devices and images following the DeepShield-Library is implemented.

This software library (based on Python) connetcs user account(s) (creator, verifier), content creator(s) (like for a device, operating system or software app), content verifier(s) (like ai, deepfake) to the blockchain network in order to initialize, read and write blockchain transactions in a tamper proof way to process images along different information platforms like social media.

# Results A.2 – FPGA based Visual Capture Device

The results of the technical demonstrator (A.2 – FPGA based Visual Capture Device) for the DeepShield approach of phase stage 1 is divided between following main components:

1. FPGA based Visual Capture Device
2. FPGA based VCD Application
3. DeepShield Integration Library

## FPGA based Visual Capture Device

The implemented FPGA-based visual capture device supports the defined requirements while the design and technical conception phase and based on elements, shown by following figure.



Figure 2: DeepShield – Module A.2 – Device

## FPGA based Visual Capture Device Application

The implemented FPGA-based visual capture device application supports the defined requirements while the design and technical conception phase and based on application views, shown by following figure to configure and use the FPGA-based visual capture device.

Several screenshots of a computer

AI-generated content may be incorrect.

Figure 2: DeepShield – Module A.2 – Application and Views

## DeepShield Integration Library

The DeepShield integration library is the key element to communicate between a FPGA based image capture device and the decentralized network provided by the routing-node. More detailed information and how to setup and use this library is described in the guideline (B\_DLT\_Guideline\_DeepShield.pdf).

# Evaluation A.2 – FPGA based Visual Capture Device

After implement of the three FPGA based Visual Capture Device they were tested and evaluated based on the pre-defined requirements and for first ground checks about Space grade readiness. All results are documented in following sub report: A2\_FFPGA\_VCD\_Evaluation.docx

# Demonstration A.2 – FPGA based Visual Capture Device

The demonstration follows the predefined requirements and the derived test cases derived. The demonstration of the FPGA based Visual Capture Device was captured as a video together with the integration in the DeepShield-Platform and the underlying Blockchain network:

A2\_FFPGA\_VCD\_Demonstration\_small.mov