Verifying Artifact Authenticity with Unclonable Optical Tags

Mónica P. Arenas[a], Gabriele Lenzini[b], Mohammadamin Rakeei[c], Peter Y. A. Ryan[d], Marjan Škrobot[e] and Maria Zhekova
SnT, University of Luxembourg, Esch-sur-Alzette, Luxembourg

Keywords: Authenticity Verification Protocol, Authentication, Integrity, Provenance, Robustness, Formal Methods.

Abstract: We study the challenge of authenticating objects. This problem is relevant when buyers need proof that a purchase is authentic and not fake. Typically, manufacturers watermark their goods, give them IDs, and provide a certificate of authenticity. Buyers, for their part, check the IDs and verify the certificate. However, even if manufacturers are honest online registration and verification are vulnerable to hacking; servers can leak private data; goods out-for-delivery can have the ID cloned and can be replaced with imitations. We propose a cyber-physical solution that combines physical properties and cryptographic protocols and that is robust against a curious registry server and attempts to physical manipulation. Security depends on two elements: (I) a material inseparably joined with an object from which we can generate digital identities and other cryptographic tokens; (ii) two novel cryptographic protocols that ensure data and object integrity and authentication of agents and objects. Besides, we show that a material with all the desired security properties exists. We can use it to coat objects, and it has optical properties, such as unclonability, from which we can build secure cryptographic protocols. We formally prove our security claims with Proverif.

1 INTRODUCTION

Combating the counterfeiting of goods protects consumers from potential harm caused by substandard products, preserves the reputation of legitimate brands, and mitigates economic losses due to parallel markets for falsified goods.

While a robust supply chain is undeniably crucial, equally important is to guarantee the integrity of products throughout their journey from manufacturers to end-users and prevent unauthorized alterations at various points along the chain. This could be obtained if the product, with all its features, were inimitable. As this is a hard requirement, standard solutions exist to add marks and seals that, themselves, cannot be reproduced or moved around easily. For example, watermarking Huang and Wu (2007) is a widely used practice that falls into this category. The security is then delegated to the technology used to produce the markings and on how hard it is to copy them with acceptable quality to fool a verifier (e.g., a human with an ultraviolet lamp).

This work combines such an idea with a *digital* cryptographic solution. It also assumes that objects can be paired with tags possessing physically unclonable features. These features are unpredictable until one extracts and processes them to create unique IDs.

Materials with peculiar unclonable and noisy features are referred to as Physical Unclonable Functions (PUFs) McGrath et al. (2019); Pappu et al. (2002). Objects of this kind are considered sources of "randomness": *natural randomness*, if the object inherently possesses it by nature, or *artificial randomness* if the randomness is created on purpose and associated with the object Voloshynovskiy et al. (2016). Finding PUF-like features on an object is uncommon McGrath et al. (2019), and where they exist, they are very object-specific and non-transferable.

Our solution stems from recent research showing that there is a material that is an optical PUF and that can be used to coat objects like a varnish Schwartz et al. (2021): thus, it is realistic to assume that objects can be enriched with natural "randomness" by being coated. However, the problem is ensuring the extraction of cryptographically secure IDs from natural "randomness" and guaranteeing that objects and coats remain physically and digitally entangled.

[a] https://orcid.org/0000-0003-0221-5032
[b] https://orcid.org/0000-0001-8229-3270
[c] https://orcid.org/0000-0002-0015-7978
[d] https://orcid.org/0000-0002-1677-9034
[e] https://orcid.org/0000-0002-7132-7591

Arenas, M., Lenzini, G., Rakeei, M., Ryan, P., Škrobot, M. and Zhekova, M.
Verifying Artifact Authenticity with Unclonable Optical Tags.
DOI: 10.5220/0012746500003767

In *Proceedings of the 21st International Conference on Security and Cryptography (SECRYPT 2024)*, pages 64-77
ISBN: 978-989-758-709-2; ISSN: 2184-7711

Object Authentication Workflow and Its Weaknesses. The authentication of objects with natural randomness requires two processes: *enrollment* and *authentication* Gao et al. (2018); Tuyls et al. (2005); Wen and Liu (2018). Enrollment happens before the object leaves the manufacturer. Its intrinsic features are acquired and processed, possibly to inject randomness to produce unpredictable IDs together with some helper data that will help retrieve exactly those IDs (the re-acquisition is a process with noise, see later). IDs and helpers are stored. Authentication is run when the object is in the hands of the consumer. Its intrinsic features are reacquired and re-processed. The helper data enables the removal of noise and the reconstruction of secret information. If the object is authentic, this ID will match exactly that stored in the database.

However, IDs and helpers must be processed, stored, and exchanged and can then be stolen, leaked, or manipulated. Additionally, when an object is in transit for delivery, it becomes vulnerable: it can be substituted with a less valuable imitation or it can have some parts (e.g., the coat or tag) tampered with. To ensure the integrity of the product and thwart any attempts to defeat the authentication phase, stringent measures must be put in place to protect against all sorts of digital and physical manipulation.

1.1 Our Contribution

Our contribution is manifold: (i) we propose two protocols, one for enrollment and one for authentication. They are secure against attackers that can tamper with both the physical object and, as a Dolev-Yao active adversary Dolev and Yao (1983), the execution of the digital protocols. (ii) We prove our security claims by model checking, and we use ProVerif Blanchet (2001) for this purpose. (iii) We suggest an architecture and spell out how to deploy both protocols in a real scenario. (iv) We *prototype* the protocols in Python and benchmark them for robustness and reliability by experimentally measuring acceptance and rejection rates on a large dataset of PUF-like samples.

We also claim that our solution rests on a set of realistic assumptions and it is user-friendly. In particular, the enrollment and authentication protocols run with a trusted device manager and an honest-but-curious registry server; the protocols require little expertise from users, simply the ability to take pictures; the cryptographic operations are carried out by a device manager for both the enrolment and authentication phases.

The security of our protocols relies on the presence of two key elements: (i) a *secure physical tag*, which is integrated into the object i.e., inseparably bound to it, and (ii) two procedures used to extract a *robust and secure fingerprint* from this tag. This fingerprint represents the digital identity of the object; which is used on cryptographic primitives together with formal proofs to ensure secure enrollment and authentication.

A further innovative aspect of our solution is that it builds on previous research, enabling us to demonstrate that a material satisfying the assumptions does indeed exist. Specifically, a material called Cholesteric Spherical Reflectors (CSRs) Geng et al. (2017); Noh et al. (2014); Schwartz et al. (2018) can be used to coat objects giving them unclonable and unique optical features that can be scanned with a microscope connected to a camera. Existing work shows that real CSR-made tags are robust and reliable Arenas et al. (2021, 2022a). *Robust* as a specific query accepts any optical response of the *same tag* despite the inevitable presence of noise and, *reliable* since the system must reject any image from *any other tag*. Other existing works show that *uniqueness* and *unpredictability* of random sources can be corrected through an error correction process based on secure fuzzy extractors Boyen (2004); Canetti et al. (2021); Dodis et al. (2004); Li et al. (2006).

2 RELATED WORK AND BACKGROUND

Reliable artifact authentication poses a formidable challenge across diverse domains, including but not limited to luxury goods, Internet of Things (IoT) devices, and more. When it comes to the protection of goods, companies actively engage in research efforts aimed at developing robust anti-counterfeiting measures and comprehensive traceability solutions throughout the supply chain Choi et al. (2013); Nam et al. (2016). Different technologies such as RFID tags and QR-codes have been devised to establish connections between physical artifacts and their corresponding digital counterparts Anandhi et al. (2020); Marktscheffel et al. (2016); Maurya and Bagchi (2018). However, these solutions fall short in important requirements, such as unclonability and tamper evidence. As an alternative, PUFs have gained prominence due to their inherent attributes of physical randomness, uniqueness, and resistance to an attacker's attempts to predict the response generated by a challenge Gassend et al. (2002); Pappu et al. (2002). Many solutions exploit these properties for key generation and authentication protocols Armknecht et al. (2009); Herder et al. (2014); Suh and Devadas (2007).

The literature suggests developing secure authentication protocols from fuzzy responses and converting them into stable keys to perform the verification process Dodis et al. (2006, 2004). Such authentication protocols typically involve a sequence of steps, such as: (i) minutiæ extraction via signal/image processing Li et al. (2017); Tuyls and Goseling (2004), (ii) error correction algorithms through robust methods (e.g., robust secure sketches), (iii) and, the derivation of secret information from stable responses (e.g., fuzzy extractors).

The entire process of obtaining a stabilized bitstring and cryptographic keys must be implemented in both phases: the *enrollment* and the *authentication*. The vendor must ensure reliable enrollment and secure communication across different channels to engender users' trust in the products to be verified. Despite all the efforts in using security tokens with unique properties, designing secure protocols is not easy to achieve as malicious entities may find ingenious ways to attack the systems.

2.1 Secure Building Blocks

The main cryptographic primitives we use to build our authenticity verification protocol include robust secure sketches and robust fuzzy extractors, as well as hash functions and digital signatures. Pairing robust secure sketches and fuzzy extractors yields a powerful error-correction process to enhance the reliability and security of identification and authentication protocols that rely on noisy sources. Such noisy sources may include biometric information (e.g., fingerprint and iris data), or in a more general context, objects that possess fingerprint-like attributes (e.g., PUFs and CSRs).

More concretely, a robust secure sketch is a mechanism that allows for the reconstruction of binary strings, despite the presence of noise. However, the bitstrings produced by the secure sketch algorithm are not uniformly distributed and hence they do not possess the necessary security requirements to be used in the context of cryptographic applications. For this purpose, a robust fuzzy extractor is used. A robust fuzzy extractor is a provably secure mechanism to generate error-tolerant, uniformly distributed random bitstrings, such as cryptographic keys and tokens, from correlated and usually noisy sources.

In particular, during the generation step, a reading of a noisy source is used to generate a random (secret) bitstring and some public data called *helper data*. At a later stage, a second, usually noisy, reading of the same source is used in conjunction with the helper data to reproduce the same secret. Robust fuzzy extractors guarantee that: 1. the helper data reveals no information about the derived secret, 2. the same secret can be reproduced, as long as the difference (noise) between the two readings is smaller than a pre-defined threshold δ and, 3. any deliberate modification of the helper data by a malicious actor will be detected. Robust fuzzy extractors utilize robust secure sketches to reliably reproduce secret values and hence we briefly describe both primitives below in more detail. Digital signatures and hash functions are used to achieve authentication of participating entities in a scenario where both symmetric and asymmetric setups (e.g., via Public Key Infrastructure) are assumed.

2.1.1 Robust Secure Sketch

A secure sketch is, in essence, an error correction technique. It consists of pair of procedures $(\mathsf{SS}, \mathsf{Rec})$ Dodis et al. (2004), standing respectively for *sketch construction* and *reconstruction*. A secure sketch assumes the existence of a function that processes n different takes of the same noisy source and outputs a single vector ω of k integer coordinates. Denoted as:

$$\omega \leftarrow \mathsf{ImgProcess}([\mathsf{take}]_1, \ldots, [\mathsf{take}]_n).$$

The SS algorithm inputs a vector ω and returns a vector of k integer elements s, called the sketch. The sketch is used in the function Rec to reconstruct the exact same ω from a noisy version of the input, say ω', if and only if the difference between them is bounded by a threshold δ. Rec should produce something unrelated to ω otherwise.

Boyen *et al.* Boyen et al. (2005) proposed a *robust secure sketch* scheme $(\mathsf{RobustSS}, \mathsf{RobustRec})$, designed to resist tampering attempts on the sketch vector s, in which the $\mathsf{RobustSS}$ algorithm also computes the digest h of the robust template, yielding as outputs h together with s. Denoted by:

$$(s,h) \leftarrow \mathsf{RobustSS}(\omega)$$

This allows one to verify in the reconstruction algorithm $\mathsf{RobustRec}$, whether the sketch s has been deliberately modified, by recomputing the digest and checking whether it matches with the digest value computed in the $\mathsf{RobustSS}$ function. We describe the reconstruction process by

$$\omega/\perp \leftarrow \mathsf{RobustRec}(\omega', s, h).$$

2.1.2 Robust Fuzzy Extractor

A robust fuzzy extractor is composed of a pair of randomized functions $(\mathsf{RobustGen}, \mathsf{RobustRep})$. The *Generate* procedure $\mathsf{RobustGen}$ requires as input a robust template ω. This functions triggers the robust secure sketch $\mathsf{RobustSS}(\omega)$ to generate a sketch s and

a digest h. The output of the RobustGen function is a random secret string R and the *public helper data* $P = (s, h, r)$, where r is a random value. We write this as:

$$(R, P) \leftarrow \mathsf{RobustGen}(\omega).$$

On the other hand, the *Reproduce* function RobustRep takes as inputs a potentially different (noisy) robust template ω' and P and reproduces the secret R', using internally the RobustRec function of the secure sketch scheme. In other words:

$$R' \leftarrow \mathsf{RobustRep}(\omega', P)$$

The two values R and R' will match if and only if the distance between ω and ω' are within a threshold δ. A fuzzy extractor scheme allows obtaining non-reversible bitstrings, which can be used as a seed to generate cryptographic keys in either symmetric or asymmetric setup Boyen et al. (2005); Shariati et al. (2012).

3 ARCHITECTURE AND THREAT MODEL

We will start by describing the overall architecture of the artifact authenticity service, and then describe a suitable threat model and discuss some of the potential attacks.

3.1 Authenticity Service and Functionality

As we aim at an architecture for artifact authenticity, we need to guarantee security, efficiency, scalability, and reliability by proposing a system involving three primary entities —with different security assumptions: a vendor (or a manufacturer), a registry server, and a user.

The process begins with a vendor. Once subscribed to a service (referred to as a legitimate vendor), the vendor is allowed to register artifacts along with specific identifying features (some of which must be unique). Note that artifact authenticity service may be established for a single vendor or may serve a consortium of vendors and manufacturers.

A registry server holds access to a database that stores artifact records. These records encompass a unique identifier (e.g., serial numbers, QR-codes), registering vendor data (e.g., vendor ID, enrolment signature), verification data, and potentially textual description of specific identifying features (dimensions, color, weight, defects, etc.) and/or video records of the registered artifact. The database may be instantiated using a decentralized private or consortium ledger, ensuring transparency and integrity of the authenticity verification process. Each legitimate vendor must register an artifact through an enrollment procedure, which can be a simple software solution or might involve specialized hardware for added security and efficacy.

Finally, a user interacts with the authenticity verification system via an object authentication protocol intending to verify the authenticity of a given object. Similarly to the enrolment procedure, authentication may involve specific hardware solutions. We should keep protocol efficient and with very few message rounds for all protocol entities. This is necessary as the artifact authenticity service should be able to handle large volumes of objects, users, and vendors.

3.2 Our Threat Model

We consider five distinct threat models: 1) a malicious vendor aiming to poison the record database; 2) a user attempting to trick the authentication protocol into returning "authenticated" when an object is fake; 3) an honest-but-curious server whose aims to learn as much information as possible about underlying secret values used to accommodate verification; 4) any malicious entity with physical access to an object (during the lifetime of the object) to tamper with a physical tag embedded in the object. This includes malicious vendors, users, or anyone else who interacts with the object; 5) any other external attacker.

The primary objective of a malicious vendor may be to exploit the authenticity service by registering counterfeit items to obtain financial gain. Another objective of such an attacker may involve completely disrupting the authenticity service by compromising its reliability and robustness. Therefore, the authenticity service should implement robust authentication mechanisms to prevent unauthorized enrollments. Conversely, a malicious vendor may seek to undermine a competing vendor (or manufacturer) by impersonating them and attempting to register an item of inferior quality as genuine. To thwart such attacks, non-repudiation, and accountability of the enrollment procedure must be ensured. A malicious user may act as a seller intending to deceive a buyer into purchasing a counterfeit object. Another potential attack that the user can launch is to downgrade the value of the object to conceal its true worth. Physical security measures must be implemented to guard against tampering and provide evidence of interference. Successful tampering would enable a malicious vendor or user to easily accomplish their objectives. Finally, an honest-but-curious server may be compromised and

exploited to launch one of the aforementioned attacks. Looking ahead, to reduce attack surfaces and simplify the security model we introduce a trusted entity referred to as a device manager (that can be implemented as a hardware solution) acting as a bridge between vendors or users on one side and the registry server on the other. Considering these potential attacks, we formally modelled and analyzed our protocol in Section 5.

4 OBJECT AUTHENTICATION PROTOCOL

Our full process for object authentication consists of two protocols: *enrollment* and *authentication*. The reference data from a physical object is collected during the *enrollment* and the *authentication* is determined whether a queried response aligns with the enrolled reference data. We start with the assumptions defining the context where the protocols work.

4.1 Security Assumptions

The security of our authentication protocols relies on certain assumptions about the objects intended for authentication. As anticipated in Section 1, we work with objects with artificial randomness that is, objects that are physically bonded with "*seals*" that possess qualities typical of an optical PUF, such as randomness, uniqueness, and tamper evidence.

That said, we will have to assume the existence of several reliable and secure procedures. Given a set of possible magnified images of an object's seal, which we indicate with $[\texttt{seal}]_1, \ldots, [\texttt{seal}]_n$, we assume that by image processing is possible to generate a unique identifying-the-object bitstring of fixed length ω despite physical and digital "noise". This processing is abstractly represented by the ImgProcess function.

Besides, we assume two functions, RobustGen and RobustRep, that satisfy security properties coming from a fuzzy extractor scheme, as described in Section 2.1. Here, (R,P) represent the private and public bitstrings, respectively, generated by applying the robust fuzzy extractor function RobustGen to ω. R should be *reliably* reconstructed if and only if we process another set of images $\{[\texttt{seal}']_i\}_{i\in I}$ taken from the same object's seal. Here, "reliably" means despite the inevitable noise in the new images, a noise usually due to environmental conditions, such as alignment, background lights, and other distortion, and noise due to the device manager used to take pictures.

In a nutshell, the processing of this set of new images results in a new identifying-the-object bitstring ω', which in turn generates a new secret bitstring R' through the RobustRep function. The process of verifying the authenticity of the object is equivalent to checking whether the reproduced R' matches the initial secret bitstring R, which solely relies on how close ω and ω' are with respect to a certain metric. However, because R and R' are private data, they cannot be directly compared and therefore, we use their corresponding digests for comparison. Formally, this requirement can be expressed as follows:

$$(R,P) \leftarrow \mathsf{RobustGen}(\omega) \text{ and}$$
$$R' \leftarrow \mathsf{RobustRep}(\omega', P)$$

then we have:

$$\mathsf{Hash}(R) = \mathsf{Hash}(R') \text{ iff } \mathsf{dist}\big(\omega,\omega'\big) \leq \delta \qquad (1)$$

where dist is a distance over the (metric) vector space of the range of ImgProcess, and δ is the tolerated bounded error. The chosen metric space is the Chebyshev distance, where $\mathsf{dist}(\omega,\omega') = \max_i\{|\omega_i - \omega'_i|\}$. Property (1) should not be satisfied when ImgProcess is applied to a set of images taken from anything else that is not the seal that was used to generate the initial (R,P) pair.

Satisfying property (1), requires carefully designed and crafted ImgProcess, RobustGen, and RobustRep functions that require to be theoretically specified and experimentally tuned to work with real objects.

That stated, we also assume that our "seal" is physically unclonable whereas the bond object-seal is tamper-evident: it cannot be broken without compromising the integrity of the seal and in particular the validity of property (1). In the remainder, we assume the availability of such seals and of functions ImgProcess, RobustGen, and RobustRep that satisfy property (1).

4.2 Feasibility of the Assumptions

Although this paper primarily discusses the security analysis of an object authentication protocol, we briefly describe a use-case material that presents unclonable properties. This material is based on liquid crystals, which are commonly found under the screens of our electronic devices. Once liquid crystals are molded in a spherical shape, they are referred to as CSRs. When mixed with a polymer, one can create a thin varnish-like spot-shaped layer over an object. The layer cannot be removed (e.g., be scratched out) without ruining the internal crystalline structure of the CSRs. In addition to being non-removable, CSRs exhibit optical properties that depend on structural variations inherent to the manufacturing process, such

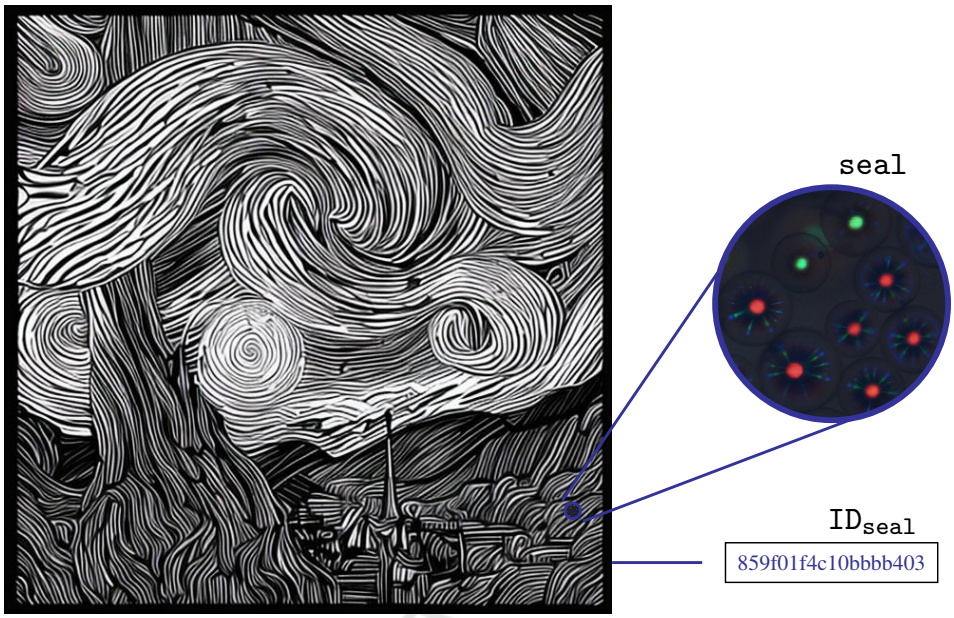

Figure 1: An illustrative example of an artifact, a CSR tag and its ID. In reality, the tag should be applied on the back of the canvas. The magnified part shows the pattern observable from the illuminated CSR. It consists of small green dots. Their exact position, color, and number are unpredictable before the first observation.

as diameter, thickness, color, and relative arrangement Geng et al. (2017, 2016); Schwartz et al. (2018). The interactions between the incident light and CSR arrays render colourful patterns (see Figure 1) that depend on the properties of CSRs, the lighting, and read-out conditions. CSRs have intrinsic unclonable properties, making them useful in security applications, including object authentication, anti-counterfeit technologies, autonomous mobile robots, among others Arenas et al. (2022b); Schwartz et al. (2021).

Thus, we assume to use "seals" made of CSRs which we can apply to objects, as shown in Figure 1. The figure shows a piece of artwork with a special CSR-made seal and a unique serial number, the former reflects lights producing unique optical patterns, while the latter is used for identification.

4.3 Requirements and Security Goals

As far as we know there are no standard functional requirements for an object authentication procedure in addition to those about security that we are going to introduce in the next paragraphs.

We assume that a vendor first registers a valuable asset. This bootstrapping procedure assumes that anyone can verify whether an object in their possession is the same as the one registered. Both, the vendor and the user, are expected to have a device manager (to acquire pictures), which is considered a trustworthy entity. We ensure that all the sensitive information is processed on the device manager to produce a secure token as a result of the image processing and a series of cryptographic operations. This secure token is associated with a unique `seal` ID ($\texttt{ID}_{\texttt{seal}}$) generated by the registry server. Thus, once private information has been processed, it is immediately discarded. Authentication should be more efficient than enrollment as the former is expected to be executed a larger number of times while enrollment only once.

With this in mind, protocols should remain capable of differentiating between authentic and counterfeit artifacts. They have been designed to satisfy the following security properties, described informally here and formalized and proven in Section 5.

- ***Uniqueness at Enrollment.*** Each artifact carrying a `seal`, at registration, is given a unique $\texttt{ID}_{\texttt{seal}}$. This means that the process of enrollment is reliable and no false-positive can occur.
- ***Reliability at Authentication.*** Two different `seal`-embedded artifacts are never recognized as equal even if one attempts to authenticate them using the same $\texttt{ID}_{\texttt{seal}}$. This means that the entropy between the features extracted from the `seal` image is high enough to avoid false positives.
- ***Robustness at Authentication.*** An authentic artifact cannot be seen as a counterfeit and no false negatives can arise.

4.4 Protocols' Agents and Principals

We assume four entities that participate in the protocol, namely the *vendor*, the *user*, the *device manager*, and *registry server*, with the following roles and responsibilities:

- The **Vendor (V)** has the right to enroll `seal` tags via a device manager. It can be a legitimate or a malicious vendor.
- The **User (U)** initiates the authenticity verification process by providing to the device manager an $\mathtt{ID}'_{\mathtt{seal}}$ and a $\mathtt{seal}'$ tag.
- The **Device Manager (DM)** is responsible for processing the images, extracting robust features, running the cryptographic primitives, and carrying out enrollment and authentication procedures.
  - During the enrollment, the DM acquires a set of $\{[\mathtt{seal}]_i\}$ images from a physical (`seal`) tag provided by the vendor. The vendor must have certain permissions to start the enrollment phase.
  - During the authentication, the user provides an $\mathtt{ID}_{\mathtt{seal}}$ to the DM, which queries the server for retrieving the public data linked to the claimed `seal`. Then, it initiates the authentication process by acquiring a set of $\{[\mathtt{seal}']_i\}$ images.

  The DM is equipped with a pair of cryptographic signing keys $(ssk_{\mathrm{DM}}, psk_{\mathrm{DM}})$. It is assumed to be a trusted entity.
- The **Registry Server (RS)** responds to the enrollment and authentication requests, maintains the database and it is also responsible for generating the $\mathtt{ID}_{\mathtt{seal}}$. It is responsible for securely storing the data during the enrollment and for retrieving the query information requested during the authentication phase. The RS is equipped with a pair of cryptographic signing keys $(ssk_{\mathrm{RS}}, psk_{\mathrm{RS}})$. It is assumed to be an honest-but-curious entity.

4.5 Enrollment Phase

The enrollment phase of the protocol involves a registered vendor V, a device manager, and a registry server, as shown in Figure 2. A vendor has an artifact and initiates the enrollment process by submitting a set of noisy images to the DM. Upon receiving the images, DM generates a nonce n_{DM}, which is sent along with a 'request' message and the vendor ID ($\mathsf{V_i}$) to the registry server. After receiving the message, the RS generates on its turn a nonce $\mathsf{n_{RS}}$ and a unique and fresh $\mathtt{ID}_{\mathtt{seal}}$. The server uses its secret signing key ($\mathsf{ssk_{RS}}$) to sign a tuple containing the vendor identification $\mathsf{V_i}$, the $\mathtt{ID}_{\mathtt{seal}}$, and concatenated nonces $\mathsf{n_{DM}}||\mathsf{n_{RS}}$. The tuple and signature are then sent back to the DM.

Upon receiving the response from the registry server, the device manager firstly verifies the message's structure and signature. If the verification is successful, it proceeds to extract a robust template ω from a set of $\{[\mathtt{seal}]_i\}$ images using the ImgProcess function. Next, it runs the robust fuzzy extractor RobustGen procedure and generates a hash value ($\mathsf{h}_{\mathtt{seal}}$) using a hash function that produces a digest of at least 256-bits. Afterward, it signs the message $\mathsf{m_{DM}}$ with its secret signing key ($\mathsf{ssk_{DM}}$) and communicates with the RS to send the message $\mathsf{m_{DM}}$ and the signature σ_{DM}. The device manager discards all the processed information.

When the registry server receives the message, it first checks the message's structure and signature. If everything is valid, the server checks whether its database already contains the helper data P and the $\mathsf{h}_{\mathtt{seal}}$.[1] If the signature verification and duplicate validation checks are successful, the RS's database stores the public information $(\mathsf{Vi}, \mathtt{ID}_{\mathtt{seal}}, \mathsf{P}, \mathsf{h}_{\mathtt{seal}})$, and the database is updated. Finally, the vendor receives a confirmation of successful enrollment of a `seal` with its respective $\mathtt{ID}_{\mathtt{seal}}$.

4.6 Authentication Phase

The authenticity verification phase involves a user `U`, a device manager, and a registry server, as shown in Figure 3. The user initiates the authenticity verification process by submitting to the DM a set of fresh images $[\mathtt{seal}']_i$ along with the identifier $\mathtt{ID}_{\mathtt{seal}}$ and $\mathsf{V_i}$ (encoded in the object). The DM generates immediately a fresh nonce $\mathsf{n_{DM}}$ and sends it along with the $\mathtt{ID}_{\mathtt{seal}}$ to the registry server.

In turn, the server generates a nonce $\mathsf{n_{RS}}$ and queries the database with the $\mathtt{ID}_{\mathtt{seal}}$. If the entry exists, DM generates the message $\mathsf{m}_{\mathtt{seal}}$ containing $\mathsf{V_i}$, $\mathtt{ID}_{\mathtt{seal}}$, $\mathsf{P}, \mathsf{h}_{\mathtt{seal}}$, and $\mathsf{n_{DM}}||\mathsf{n_{RS}}$. The nonces $\mathsf{n_{DM}}$ and $\mathsf{n_{RS}}$ are added to prevent replay attacks. The RS signs the message with its secret signing key ($\mathsf{ssk_{RS}}$).

The DM receives the message $\mathsf{m}_{\mathtt{seal}}$ together with the signature $\sigma_{\mathtt{seal}}$ and it first checks whether V'_i matches V_i. If so, the device manager verifies the signature, runs the image processing algorithm, and extracts the robust template ω'. The reproduction procedure RobustRep is executed by using the robust template ω' and the helper data P as inputs, and it outputs the secret R'. Next, the device manager gener-

[1] Note that one can employ data structures from Grossi and Vitter (2000) to facilitate highly efficient duplicate search algorithms.

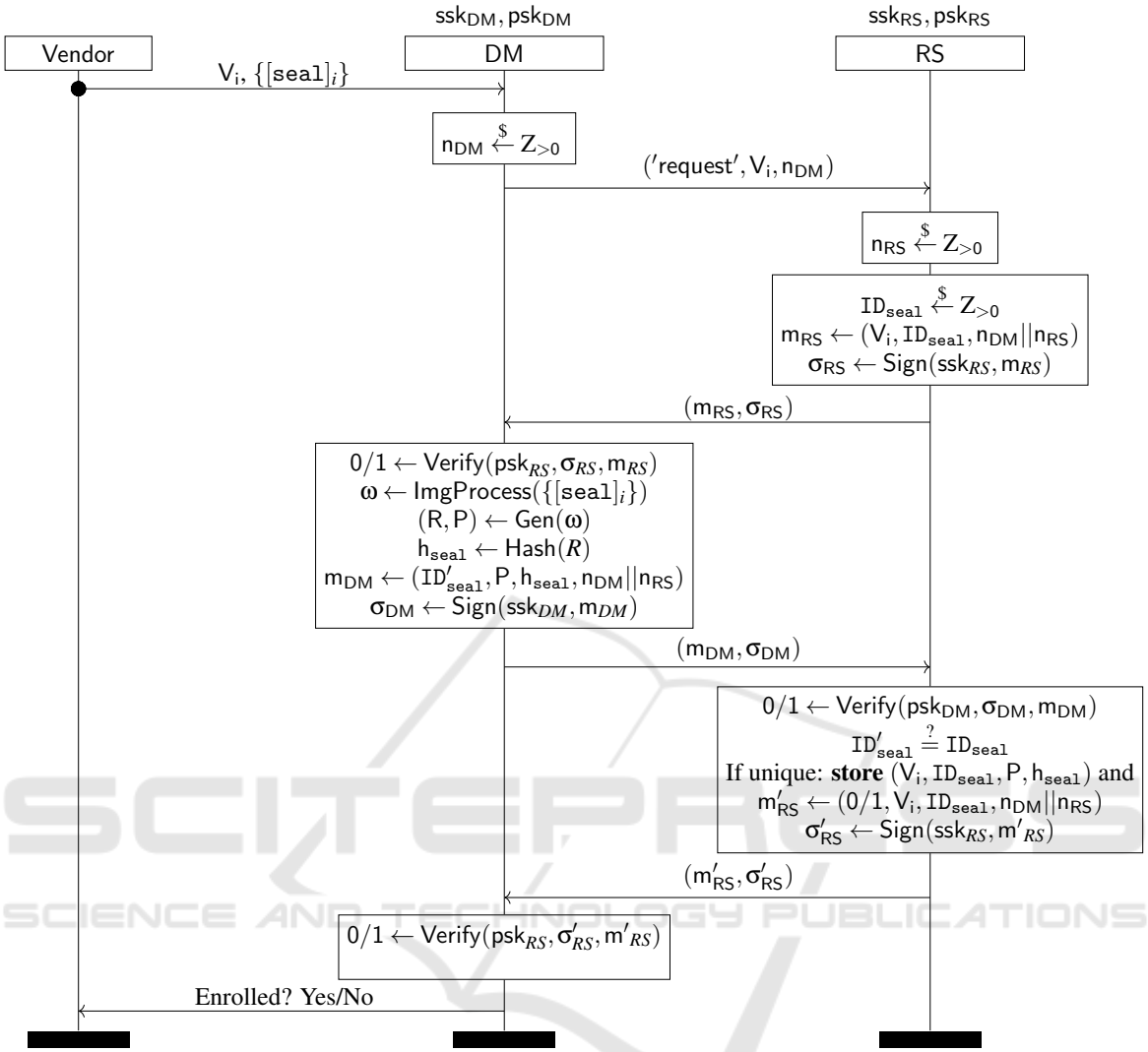

Figure 2: Message sequence chart of the enrollment phase. An arrow $\rightarrow$ is a public channel; an arrow $\bullet\!\!\rightarrow$ is a private channel, i.e., the parties have agreed on a session key, e.g., via Transport Layer Security (TLS).

ates a hash value $h'_{\mathtt{seal}}$ by using a hash function with the seed being the secret value R'. Finally, the device manager compares the new hash value with the one received from the registry server ($h_{\mathtt{seal}} \stackrel{?}{=} h'_{\mathtt{seal}}$). A successful comparison confirms the genuineness of the artifact and the user receives a confirmation of successful authentication.

5 SECURITY ANALYSIS

We will discuss security against specific adversaries and adversarial models and we start with the definitions of the adversary types.

Definition 1. *A **semi-honest adversary**, or honest-but-curious adversary Goldreich (2009), is a type of attacker who can perform any passive attack on a system and can gather as much data as possible without breaking the prescribed definition of the protocol.*

Definition 2. *A **malicious adversary** is the strongest type of adversary Goldreich (2009). Such an adversary can deviate from the protocol's definition and use any effective strategy to gain further knowledge about private data belonging to other parties or manipulate the computation's outcome.*

5.1 Adversary Model

We assume a Dolev-Yao adversary Dolev and Yao (1983), a standard choice in security protocol analysis. It represents an external active intruder that controls the communication channels between the other

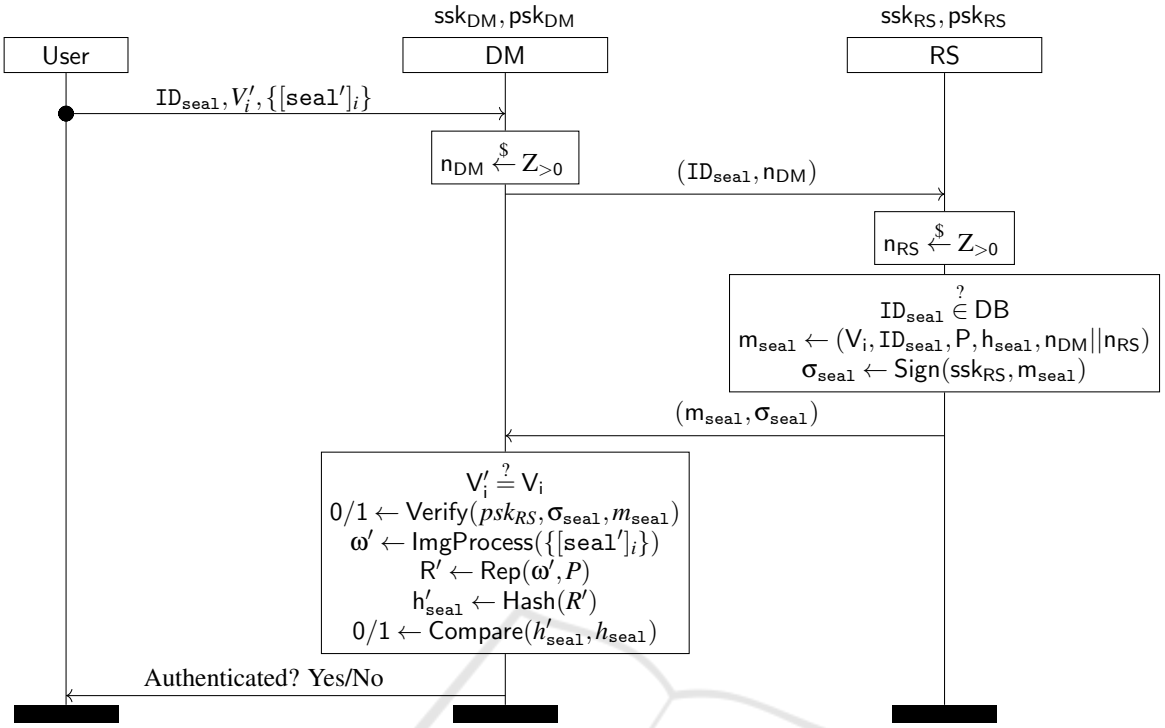

Figure 3: Message sequence chart of the authenticity verification phase. An arrow $\rightarrow$ is a public channel; an arrow $\bullet\!\!\rightarrow$ is a private channel, i.e., the parties have agreed on a session key, e.g., via TLS.

principals: the intruder can intercept, modify, create/inject, and destroy messages, but he cannot create signatures unless he knows the appropriate keys.

We also assume that our adversary can manipulate objects. Although it has already been stated that tampering with the "seal" is not possible without compromising its integrity, the intruder can still modify the ID that an object o carries together with its seal, something we write as $o[\mathtt{seal}, \mathtt{ID}_{\mathtt{seal}}]$. In short, the intruder can change $\mathtt{ID}_{\mathtt{seal}}$: $\mathsf{TamperID}(o[\mathtt{seal}, \mathtt{ID}_{\mathtt{seal}}]) \rightarrow o[\mathtt{seal}, \mathtt{ID}'_{\mathtt{seal}}]$.

Instead, we assume that the adversary cannot tamper with the device manager. That said, we also assume that the adversary can act as a *Vendor*, a *User*, or an external attacker with goals of: (i) learning sensitive parameters: $[\mathtt{seal}]_i$, ω, and R; (ii) enrolling the object at the RS by bypassing DM; (iii) impersonating the RS in the enrollment phase; and (iv) succeeding to authenticate an object without previously being enrolled.

Instead, we assume that the registry server is honest-but-curious which means it obeys the protocol but is trying to obtain knowledge about $[\mathtt{seal}]_i$, ω, and R. Moreover, in our protocol, we assume that the communication channel between the *vendor* and the DM, as well as the user and the DM, are private. We also assume that the secret information (e.g., ω, R) extracted from each `seal` is discarded once the message is digitally signed. This requires that we trust the device manager to be stateless.

5.2 Formal Analysis by ProVerif

We formally prove that our design satisfies the protocol's main goal, as we stated in Section 4.3. We check the security of our protocol using ProVerif model-checker Blanchet (2001). ProVerif is a formal verification tool that can prove the secrecy, authentication, and privacy properties of a protocol and supports an unbounded number of messages and sessions. It takes as input the description of a protocol in a cryptography-enhanced extension of the applied π-calculus Abadi et al. (2017) and automatically translates this description into Horn clauses, which is used to prove the desired properties in the Dolev-Yao adversarial model. ProVerif is sound, which means that if it proves a certain property holds, then this property is satisfied.

Cryptographic primitives in ProVerif are modeled using equational theories. Table 1 shows the equational theories used in our model. The security requirements that we aim to achieve in our protocol are secrecy and authentication properties. ProVerif verifies the secrecy of a parameter by querying the reachability of all possible traces in the protocol to

find a trace where the target secret is known to the attacker. To prove authentication properties, the verifier uses correspondence assertions on events. Events in ProVerif are internal messages, not accessible to the attacker, that flag the protocol state within a trace without affecting the protocol behavior. They may contain arguments as inputs to precisely capture a state inside a process.

Table 1: Equational theories in our protocol model.

Primitive	Equational Theory
Signature	$\mathsf{getmess}(\mathsf{sign}(m, ssk)) = m$ $\mathsf{checksign}(\mathsf{sign}(m, ssk), pk(ssk)) = m$
Reproduction	$\mathsf{rep}(\omega, \mathsf{genP}(\omega)) = \mathsf{genR}(\omega)$

The following list of events is used in our ProVerif model:

- **serverChallenged**: raised by the RS upon receiving a request from the DM.
- **serverVerified**: raised by the DM after successfully verifying the RS's signature on the message m_{RS}.
- **enrolRequested**$(V_i, ID_{seal}, P, h_{seal})$: raised by the DM when it sends the enrollment request for an object with (ID_{seal}, P, h_{seal}) parameters issued by the vendor's identity, V_i.
- **enrolVerified**$(V_i, ID_{seal}, P, h_{seal})$: raised by the RS when it successfully verifies the signature of the DM on the message m_{DM}.
- **resultVerified**(V_i, ID_{seal}): raised by the DM when it successfully verifies the signature of the RS on the message m'_{RS}.
- **recordSent**$(V_i, ID_{seal}, P, h_{seal})$: raised by the RS when it sends the message, m_{seal}.
- **recordVerified**$(V_i, ID_{seal}, P, h_{seal})$ raised by the DM when the validity checks for h_{seal} and V_i are true and it successfully verifies the signature of the RS on the message m_{seal}.
- **honestVendors**$(V_i, csrH)$ This event is used to distinguish an honest vendor process, enrolling an object with csrH image, from the vendor process controlled by the attacker. This event helps ProVerif to discard false attacks during the secrecy check.

5.3 Security Properties

As we mentioned in Section 5.2, we investigate several secrecy and authentication properties in our protocol. The first step to verify the authentication properties is to formally define them in a proper mathematical expression. Here, we present a formal definition of three authentication properties and later in Section 5.4, we show the result of ProVerif analysis based on these definitions.

Definition 3 (RS Authenticity)**.** *The Registry Server Authenticity holds if each occurrence of the* `serverVerified` *event is preceded by a unique occurrence of the* `serverChallenged` *event.*

One of the key authentication requirements for our protocol is that the DM should assure that it indeed enrolls an object to a server that possesses the pair of $\{ssk_{RS}, psk_{RS}\}$ keys. The RS Authenticity guarantees that the message m_{RS}, received by the DM, is genuinely generated by the RS, hence the server is authentic for the DM. This property is injective which means that there is a one-to-one correspondence between the `serverVerified` and `serverChallenged` events. In other words, the injective constraint is used to prevent replay attacks.

$$inj\text{-}\mathtt{serverVerified} \rightsquigarrow inj\text{-}\mathtt{serverChallenged}$$

Definition 4 (Enrollment Authentication)**.** *The Enrollment Authentication is valid if each occurrence of the* $\mathtt{resultVerified}(V_i, ID_{seal})$ *event is preceded by a unique occurrence of the* $\mathtt{enrolVerified}(V_i, ID_{seal}, P, h_{seal})$ *and each occurrence of the event* $\mathtt{enrolVerified}(V_i, ID_{seal}, P, h_{seal})$ *is preceded by a unique occurrence of the* $\mathtt{enrolRequested}(V_i, ID_{seal}, P, h_{seal})$ *event.*

This important security requirement in the enrollment phase of our protocol says that only an authentic DM must be able to enroll an object to the RS and the integrity of the sent record must be preserved. In addition, it implies that when the RS returns the enrollment result to the DM, the DM makes sure that this result is genuinely generated by the RS for the same object. More precisely, the Enrollment Authentication property assures the RS that the message m_{DM} received by the RS is generated by a legitimate DM and its content $(V_i, ID_{seal}, P, h_{seal})$ has not been manipulated by the attacker, hence this record is authentic and can be stored in the database. On the other side, this mutual authentication property guarantees the DM that the message m'_{RS} is generated by the RS and contains the result of the enrollment request for the object with (V_i, ID_{seal}) information.

$$inj\text{-}\mathtt{resultVerified}\ (V_i, ID_{seal}) \rightsquigarrow (inj\text{-}\mathtt{enrolVerified}\ (V_i, ID_{seal}, P, h_{seal}) \rightsquigarrow inj\text{-}\mathtt{enrolRequested}\ (V_i, ID_{seal}, P, h_{seal}))$$

Definition 5 (Object Authentication)**.** *The Object Authentication holds if each occurrence of the* $\mathtt{recordVerified}(V_i, ID_{seal}, P, h_{seal})$ *event is preceded by a unique occurrence of the* $\mathtt{recordSent}(V_i, ID_{seal}, P, h_{seal})$ *event.*

Table 2: ProVerif results of the proposed authenticity verification protocol.

	Secrecy			Authentication		
Properties	$\{\lceil \texttt{seal} \rceil_i\}$	ω	R	Reg. Server	Enrollment	Object
	✓	✓	✓	✓	✓	✓

This property assures to the device manager that (i) The message $\mathsf{m}_{\texttt{seal}}$ is generated by the registry server and its content is not manipulated by an attacker. (ii) The hash term $\mathsf{h}_{\texttt{seal}}$ derived from the user's provided input $[\texttt{seal}']_i$ matches the one retrieved from the registry server's database. (iii) The vendor ID V_i', presented by the user is the same as the vendor ID V_i, stored in the registry server's database for the object with the identity of $\texttt{ID}_{\texttt{seal}}$.

$$inj\text{-}\texttt{recordVerified}\ (\mathsf{V}_\mathsf{i}, \texttt{ID}_{\texttt{seal}}, \mathsf{P}, \mathsf{h}_{\texttt{seal}}) \rightsquigarrow inj\text{-}\texttt{recordSent}\ (\mathsf{V}_\mathsf{i}, \texttt{ID}_{\texttt{seal}}, \mathsf{P}, \mathsf{h}_{\texttt{seal}})$$

5.4 ProVerif Results

Our ProVerif model[2] is composed by the User, Vendor, DM, and RS, with definitions provided in Section 4.4, and two types of adversaries whose power are defined in Definition 1 and 2.

Proving secrecy properties in ProVerif is a straightforward procedure and ProVerif just needs to check if a target parameter is not reachable to the attacker for all possible traces. The analysis results show that the secrecy of $[\texttt{seal}]_i$, ω, and R in both malicious and semi-honest attacker models are satisfied, as depicted in Table 2.

To prove the authentication properties in our ProVerif model, we express three RS Authenticity, Enrollment Authentication, and Object Authentication definitions as standard correspondence queries. ProVerif successfully proves that all these three properties hold in our protocol which means: (i) all enrollments are done by an authentic registry server; (ii) if an enrollment record is stored in the database then this record has been enrolled by a trusted DM and its integrity is preserved; (iii) if an authentication request for an object claimed to be produced by a vendor is successful, then this object has previously been enrolled in the authentic RS by the same vendor. This result is very important because it guarantees that an attacker cannot authenticate a fake object or manipulate the vendor's ID associated with an authentic object.

[2]ProVerif and Python codes available in https://edu.lu/u7ue8.

6 IMPLEMENTATION DISCUSSION

A proof-of-concept (POC) of the proposed protocol was implemented in Python 3.9.7[2] on a macOS Unix machine (Apple M1 Pro chip with 32 GB RAM). The protocol consists of seals made up of CSRs, which can be extended to other materials with similar noisy behavior. As demonstrated by the authors in another paper Arenas et al. (2022b), the image processing is the most computationally expensive operation in both enrollment and authentication phases. In contrast, other operations like signing, verifying, and hashing are highly efficient, taking only a fraction of a second to complete.

6.1 Hash Function

The hash function used in the secure sketch scheme $(\mathsf{RobustSS}, \mathsf{RobustRec})$ and the enrollment routine can be selected from well-known families of hash functions. These families provide resistance to preimage and collision attacks, such as the SHA-2 and SHA-3 families. In the classical setting, i.e., when no adversaries with quantum capabilities are assumed, the digest size should be at least 256-bits. In this case, the choice of SHA-256 is sufficient.

6.2 Entropy

The entropy of a biometric vector is calculated via the relation $m = n\log_2(kaN)$, where N is the length of the number line, k is the interval's length, a is a fixed constant, and n is the size of the biometric vector, as well described by Li et al. (2017). Further, the entropy heavily relies on the density of the minutiæ and their distribution on the number line. As mentioned, the use-case material is made up of CSRs, composed of spherical particles called *blobs*, where each CSR image contains arrays with different blob densities and blob diameters (k). The image processing involves the blob's extraction and the creation of the robust template ω. To create this template, the extracted blobs from $i-$images are embedded into a grid that is a product of two number lines, each of length N, in a similar way as Li et al. (2017). The entropy of the robust template ω in this instance relies on the length N

of each number line, the size of each square, as well as the number of blobs contained in the CSR image and its average blobs' diameter. Table 3 reports the minimum and maximum min-entropy values of the robust template vector ω extracted from our dataset.

Table 3: Entropy of ω.

	N	k	Num. of blobs	Min-entropy
min	15	8	5	3 108-bits
max	70	24	201	104 999-bits

6.3 Robustness and Reliability

We developed in Arenas et al. (2021, 2022a) a reliable process to extract the information from images captured under different lighting and environmental conditions, a methodology also adopted in this paper. We acquired a reference image from 30 different CSR tags and, to have a larger dataset, we generated a set of images by artificially injecting noise into each reference image. The dataset was generated by adding similarity and Gaussian noise. For the former, we randomly added rotation, illumination changes, and blurring effect (lack of focus). For the Gaussian noise, we introduced values with a standard deviation ranging from 0.0 to 0.3 —values chosen considering realistic conditions.

We assumed that the enrollment is executed in a single attempt, from which five noisy images were generated. And, for the authentication phase, 20 attempts were considered at different time intervals, while five images were also acquired from each attempt. This experimental setup ensures the extraction of robust information.

The robustness of our system was benchmarked by comparing 600 pairs of related CSR retakes and our system's reliability was evaluated by comparing 600 pairs of unrelated CSR retakes. A robust authentication process should be able to accept any image of an optical response from the same CSR despite the presence of noise during the readout. Similarly, a reliable authentication process should reject any image generated by any other CSR. The results of these comparisons are presented in Table 4. The protocol's acceptance rate for related CSR images was 93.75%, and the false-negative rate of 6.25% was due to the difference between ω and ω'. Our authentication protocol also rejected all incorrect (CSR, $\mathsf{ID}_{\mathsf{CSR}}$) pairs, resulting in a false-positive ratio of 0%, confirming the reliability of our system.

Table 4: Confusion Matrix

		Actual Value	
		Accepted	**Rejected**
Result	**Accepted**	93.75%	0%
	Rejected	6.25%	100%

7 CONCLUSION AND FUTURE WORK

We studied the problem of how to authenticate objects even in the presence of adversaries that have an interest in tampering with the physical items and the data. We proposed two procedures, one for enrollment and one for authentication. Their security features rely on physical elements that are natural PUFs (i.e., special tags we called "`seal`"), on secure information extraction algorithms, and on specific cryptographic protocols.

We are interested in securing the procedures against two sources of attacks: physical and cyber-physical. In the former, an adversary tries to physically transfer a `seal` from one artwork to another. Based on the assumption that the `seal` are tamper evident, we guarantee object integrity, making it difficult for the attacker to succeed. In the latter, the cyber-physical attack, a Dolev-Yao adversary tries to fool the communication channels between two key parties: the device manager and the registry server. Nevertheless, our procedures preserve data secrecy of sensitive data, that is, the set of images taken from the `seal`, the robust features ω, and the secret value R obtained via the fuzzy extractor primitive. They also satisfy registry server authenticity, enrollment authentication, and object authentication. All our security claims have been formally verified using the formal verification tool ProVerif.

There is still future work to do. In our analysis, we did not consider an adversary that can tamper with the process of extracting information, for instance, by corrupting the camera so that the device will return a constant value ω. This corruption can be implemented e.g., by forcing ImgProcess to work always on the same set of images injected by the adversary. In our message chart, this assumption is reflected in the interaction between the vendor or user and the device manager being a private channel, where the device is assumed trustworthy. If we admitted a device's being corrupted, that is, if we assumed the vendor-device (or user-device) channel be in clear in our model, we expect that the overall security collapses: there is, abstractly speaking, no difference between corrupting the seal or the pictures taken from the seal. An open

question for future work is how to model a large set of physical adversarial capabilities, including code corruption of a device in a way that is realistically aligned with what can be done in terms of software vulnerabilities.

ACKNOWLEDGEMENTS

The authors acknowledge the financial support from the Luxembourg National Research Fund (FNR) on the Secure and Verifiable Electronic Testing and Assessment Systems –SEVERITAS (INTER/ANR/20/14926102 ANR-20-CE39-009-03). Marjan Škrobot received support from the FNR under the CORE Junior project (C21/IS/16236053/FuturePass). Peter Y.A. Ryan received support from FNR under the CORE project (C21/IS/16221219/ ImPAKT). The authors also acknowledge Prof. Dr. J. Lagerwall and the ESMP group for providing the CSR images.

REFERENCES

Abadi, M., Blanchet, B., and Fournet, C. (2017). The applied pi calculus: Mobile values, new names, and secure communication. *Journal of the ACM (JACM)*, 65(1):1–41.

Anandhi, S., Anitha, R., and Sureshkumar, V. (2020). An Authentication Protocol to Track an Object with Multiple RFID Tags Using Cloud Computing Environment. *Wireless Personal Communications*, 113(4):2339–2361.

Arenas, M., Demirci, H., and Lenzini, G. (2021). Cholesteric Spherical Reflectors as Physical Unclonable Identifiers in Anti-counterfeiting. In *16th International Conference on ARES*, pages 1–11. ACM.

Arenas, M., Demirci, H., and Lenzini, G. (2022a). An Analysis of Cholesteric Spherical Reflector Identifiers for Object Authenticity Verification. *Machine Learning and Knowledge Extraction*, 4(1):222–239.

Arenas, M. P., Bingol, M. A., Demirci, H., Fotiadis, G., and Lenzini, G. (2022b). A Secure Authentication Protocol for Cholesteric Spherical Reflectors using Homomorphic Encryption. In *AFRICACRYPT*.

Armknecht, F., Maes, R., Sadeghi, A. R., Sunar, B., and Tuyls, P. (2009). Memory leakage-resilient encryption based on physically unclonable functions. In *International Association for Cryptologic Research*, volume 5912 LNCS, pages 685–702.

Blanchet, B. (2001). An efficient cryptographic protocol verifier based on prolog rules. In *Proc. 14th IEEE Computer Security Foundations Workshop*, volume 1, pages 82–96, Nova Scotia. Citeseer, IEEE.

Boyen, X. (2004). Reusable cryptographic fuzzy extractors. In Atluri, V., Pfitzmann, B., and McDaniel, P. D., editors, *Proceedings of the 11th CCS, 2004*, pages 82–91. ACM.

Boyen, X., Dodis, Y., Katz, J., Ostrovsky, R., and Smith, A. D. (2005). Secure remote authentication using biometric data. In Cramer, R., editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 147–163. Springer.

Canetti, R., Fuller, B., Paneth, O., Reyzin, L., and Smith, A. (2021). Reusable Fuzzy Extractors for Low-Entropy Distributions. *Journal of Cryptology*, 34(1):2.

Choi, S., Yang, B., Cheung, H., and Yang, Y. (2013). Data management of RFID-based track-and-trace anti-counterfeiting in apparel supply chain. In *8th International Conference for Internet Technology and Secured Transactions (ICITST-2013)*, pages 265–269.

Dodis, Y., Katz, J., Reyzin, L., and Smith, A. (2006). Robust fuzzy extractors and authenticated key agreement from close secrets. In Dwork, C., editor, *Advances in Cryptology - CRYPTO 2006*, pages 232–250, Berlin, Heidelberg. Springer Berlin Heidelberg.

Dodis, Y., Reyzin, L., and Smith, A. (2004). Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *International conference on the theory and applications of cryptographic techniques*, pages 523–540. Springer.

Dolev, D. and Yao, A. (1983). On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208.

Gao, Y., Su, Y., Xu, L., and Ranasinghe, D. C. (2018). Lightweight (Reverse) Fuzzy Extractor with Multiple Referenced PUF Responses. arXiv:1805.07487 [cs].

Gassend, B., Clarke, D., van Dijk, M., and Devadas, S. (2002). Controlled physical random functions. In *18th Annual Computer Security Applications Conference, 2002. Proceedings.*, pages 149–160. ISSN: 1063-9527.

Geng, Y., Noh, J., Drevensek-Olenik, I., Rupp, R., and Lagerwall, J. (2017). Elucidating the fine details of cholesteric liquid crystal shell reflection patterns. *Liquid Crystals*, 44(12-13):1948–1959.

Geng, Y., Noh, J., Drevensek-Olenik, I., Rupp, R., Lenzini, G., and Lagerwall, J. P. (2016). High-fidelity spherical cholesteric liquid crystal Bragg reflectors generating unclonable patterns for secure authentication. *Scientific Reports*, 6:1–9.

Goldreich, O. (2009). *Foundations of cryptography. 2: Basic applications*. Cambridge Univ. Press, Cambridge.

Grossi, R. and Vitter, J. S. (2000). Compressed suffix arrays and suffix trees with applications to text indexing and string matching (extended abstract). In Yao, F. F. and Luks, E. M., editors, *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 397–406. ACM.

Herder, C., Yu, M. D., Koushanfar, F., and Devadas, S. (2014). Physical unclonable functions and ap-

plications: A tutorial. *Proceedings of the IEEE*, 102(8):1126–1141.

Huang, S.-Y. and Wu, J. (2007). Optical watermarking for printed document authentication. *IEEE Transactions on Information Forensics and Security*, 2(2):164–173.

Li, N., Guo, F., Mu, Y., Susilo, W., and Nepal, S. (2017). Fuzzy extractors for biometric identification. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 667–677. IEEE.

Li, Q., Sutcu, Y., and Memon, N. (2006). Secure sketch for biometric templates. *Lecture Notes in Computer Science*, 4284 LNCS:99–113.

Marktscheffel, T., Gottschlich, W., Popp, W., Werli, P., Fink, S. D., Bilzhause, A., and de Meer, H. (2016). QR code based mutual authentication protocol for Internet of Things. In *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6.

Maurya, P. K. and Bagchi, S. (2018). A Secure PUF-Based Unilateral Authentication Scheme for RFID System. *Wireless Personal Communications*, 103(2):1699–1712.

McGrath, T., Bagci, I. E., Wang, Z. M., Roedig, U., and Young, R. J. (2019). A PUF taxonomy. *Applied Physics Reviews*, 6(1):011303–(1–25).

Nam, H., Song, K., Ha, D., and Kim, T. (2016). Inkjet Printing Based Mono-layered Photonic Crystal Patterning for Anti-counterfeiting Structural Colors. *Scientific Reports*, 6(1).

Noh, J., Liang, H.-L., Drevensek-Olenik, I., and Lagerwall, J. P. (2014). Tuneable multicoloured patterns from photonic cross-communication between cholesteric liquid crystal droplets. *Journal of Materials Chemistry C*, 2(5):806–810.

Pappu, R., Recht, B., Taylor, J., and Gershenfeld, N. (2002). Physical One-Way Functions. *Science*, 297(September):2026–2031.

Schwartz, M., Geng, Y., Agha, H., Kizhakidathazhath, R., Liu, D., Lenzini, G., and Lagerwall, J. P. F. (2021). Linking Physical Objects to Their Digital Twins via Fiducial Markers Designed for Invisibility to Humans. *Multifunctional Materials*.

Schwartz, M., Lenzini, G., Geng, Y., Rønne, P. B., Ryan, P. Y., and Lagerwall, J. P. (2018). Cholesteric Liquid Crystal Shells as Enabling Material for Information-Rich Design and Architecture. *Advanced Materials*, 30(30):1–19.

Shariati, S., Standaert, F. X., Jacques, L., and Macq, B. (2012). Analysis and experimental evaluation of image-based PUFs. *Journal of Cryptographic Engineering*, 2(3):189–206.

Suh, G. E. and Devadas, S. (2007). Physical Unclonable Functions for Device Authentication and Secret Key Generation. In *2007 44th ACM/IEEE Design Automation Conference*, pages 9–14. ISSN: 0738-100X.

Tuyls, P., Akkermans, A. H., Kevenaar, T. A., Schrijen, G. J., Bazen, A. M., and Veldhuis, R. N. (2005). Practical biometric authentication with template protection. *Lecture Notes in Computer Science*, 3546:436–446.

Tuyls, P. and Goseling, J. (2004). Capacity and examples of template-protecting biometric authentication systems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3087:158–170.

Voloshynovskiy, S., Holotyak, T., and Bas, P. (2016). Physical object authentication: Detection-theoretic comparison of natural and artificial randomness. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2029–2033, Shanghai. IEEE.

Wen, Y. and Liu, S. (2018). Robustly Reusable Fuzzy Extractor from Standard Assumptions. In Peyrin, T. and Galbraith, S., editors, *Advances in Cryptology – ASIACRYPT 2018*, volume 11274, pages 459–489. Springer International Publishing, Cham. Series Title: Lecture Notes in Computer Science.