Computers & Security 148 (2025) 104131

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Remote secure object authentication: Secure sketches, fuzzy extractors, and security protocols

Mónica P. Arenas, Georgios Fotiadis, Gabriele Lenzini *, Mohammadamin Rakeei

SnT, University of Luxembourg, Esch-sur-Alzette, Luxembourg

ARTICLE INFO

Keywords:
Authentication
Secrecy
Remote authentication protocol
Robust secure sketch
Robust fuzzy extractor
Fingerprint-like features

ABSTRACT

Coating objects with microscopic droplets of liquid crystals makes it possible to identify and authenticate objects as if they had biometric-like features: this is extremely valuable as an anti-counterfeiting measure. How to extract features from images has been studied elsewhere, but exchanging data about features is not enough if we wish to build secure cryptographic authentication protocols. What we need are authentication tokens (i.e., bitstrings), strategies to cope with noise, always present when processing images, and solutions to protect the original features so that it is impossible to reproduce them from the tokens. Secure sketches and fuzzy extractors are the cryptographic toolkits that offer these functionalities, but they must be instantiated to work with the peculiar specific features extracted from images of liquid crystals. We show how this can work and how we can obtain uniform, error-tolerant, and random strings, and how they are used to authenticate liquid crystal coated objects. Our protocol reminds an existing biometric-based protocol, but only apparently. Using the original protocol as-it-is would make the process vulnerable to an attack that exploits certain physical peculiarities of our liquid crystal coatings. Instead, our protocol is robust against the attack. We prove all our security claims formally, by modeling and verifying in Proverif, our protocol and its cryptographic schemes. We implement and benchmark our solution, measuring both the performance and the quality of authentication.

1. Introduction

Microscopic spheres of cholesteric liquid crystals, the same liquid used in LCD screens, can be used today to give an object a unique identity (Geng et al., 2016; Lenzini et al., 2017). An object selectively coated with a layer of crystal can reflect light, generating patterns that are not predictable before one observes them for the first time with a microscope (see Fig. 1). An ensemble of spheres – called Cholesteric Spherical Reflectors (CSRs)– is physically unclonable: it is not possible to predict where the spheres are laying on a surface (their position depends on uncontrollable factors during the production phase), and it is unfeasible to foresee, at least in full detail, how the colored pattern will look like since it depends on how light is reflected across all the spheres and the medium hosting them.

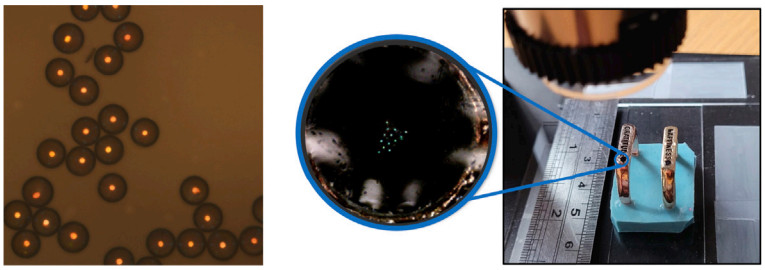
CSRs have great potential in security applications (Lenzini et al., 2017; Schwartz et al., 2021). An obvious application is to use CSRs in anti-counterfeiting. They have a wide applicability because they can be used to coat several types of objects e.g., goods of any kind like jewelry, any sort of papers like packages and documents, edible items like drugs and food. CSRs could be a game-changer technology in remote authentication, but first we need to solve several non-trivial challenges. The way to interact with CSRs is to take a picture of their generated patterns. Thus, one challenge is to extract identifying features from CSR images. Arenas et al. (2021, 2022b) show that almost circular spots of colors are always present in practically all CSR's image (they call them "blobs", see Fig. 2). Such elements are *minutiæ*, specific features that contain enough information to discern one image from another. Arenas et al. model blobs as arrays of colored circles (i.e., coordinate of the center, radius, and average color in RGB). In that multi-dimensional metric, using the information carried by the circles, they show that it is possible to re-identify with high-reliability an image and to distinguish it from other CSR images.

But if one wished to extract cryptographically secure bitstrings from an image, the task is made hard by the presence of noise, which introduces further challenges that recall those in biometrics authentication:

(i) the extracted features need to be transformed into fixed-length binary strings. Besides, the bitstrings have to be *robust*, i.e., tolerant to noise at bit-level, and *non-invertible* (Nandakumar and Jain, 2015a) i.e., protecting the features from spoof and replay attacks;

* Corresponding author.
E-mail address: gabriele.lenzini@uni.lu (G. Lenzini).

https://doi.org/10.1016/j.cose.2024.104131
Received 10 April 2023; Received in revised form 15 September 2024; Accepted 19 September 2024
Available online 27 September 2024



(a) (b)

Fig. 1. CSR responses taken with a: (a) professional polarized microscope, (b) USB DinoLite microscope –CSRs engraved in a ring.

(ii) authentication needs to be achieved by cryptographically secure protocols, because CSRs are mainly meant to authenticate objects from remote (e.g., to verify that a purchased diamond is authentic).

The problem of developing secure object authentication protocols from CSRs shares similarities with biometric authentication protocols. Thus, a first idea could be to resort to solutions that are analogous to those available for biometrics authentication.

For instance, we could simply take the "blobs" (i.e., the minutiæ extracted from images of CSR), encrypt them, register and store them encrypted, and then use homomorphic encryption to check whether the features of an object we wish to authenticate have been previously registered. Homomorphic encryption is a powerful tool for performing operations on sensitive data in the encrypted domain, hence ensuring the privacy of sensitive information. In fact, this idea has been proposed (Arenas et al., 2022a), but performing homomorphic encryption operations requires a substantial computational effort. We prefer lightweight solutions that can be carried out by resource-constrained devices, like mobile phones.

Thus, in this paper, we follow a different approach: we use *secure sketches* and *fuzzy extractors*, originally introduced in Dodis et al. (2004) and then use their output in an authentication protocol specifically designed to work with CSR. Secure sketches are used for extracting *sequences of bits* from noisy ("fuzzy") sources; the bits are reproducible despite intra-subject variations in the source. Fuzzy extractors (Boyen et al., 2005; Shariati et al., 2012) are used to obtain non-reversible and cryptographically secure bitstrings. Secure sketches and fuzzy extractors can offer an efficient solution for developing secure authentication protocols from fuzzy inputs, since the operations they perform are usually simple operations on bitstrings (e.g., hash operations and integer additions/subtractions). We assume to have access to extract features from CSR images. This step has been proven elsewhere (Arenas et al., 2022b).

Contribution. More concretely, this work proposes and implements solutions for both challenges (i) and (ii) described above. 1. We proposed a two-dimensional space (number grid) to obtain a *fixed-length array* where the robust features are embedded, ensuring invariability in the matrix size across different images of the same CSR (i.e., intra-subject variations); 2. we have extended the authentication protocol proposed in Li et al. (2017), tailored it to CSRs and proved that secrecy and security properties are achieved. We also proved that our enrollment and authentication phases are secure against a Dolev–Yao adversary and curious servers; 3. we provided a proof-of-concept implementation of an authentication protocol using the CSR technology in combination with secure sketches and fuzzy extractors. We additionally benchmarked the computational efficiency and quality of authentication.

2. Background and related work

Several sources of noise provide similar but not identical readings at each read-out and from which one can potentially generate high entropy information uniformly distributed and not easily reproducible. These sources include biometric systems – such as fingerprints, faces, and irises – physical sources like Physical Unclonable Functions (PUFs), and quantum devices. This work relates to the first two.

To extract information from these physical sources and to put them in the context of cryptographic applications, Dodis et al. (2004) proposed a secure sketch and fuzzy extractor model which can reliably extract a randomness R from a biometric input ω. Even if the input changes, the same R can be generated in a certain fault tolerance range. However, their proposed model is vulnerable to passive attacks. Thus, Boyen et al. (2005) offered some improvements to the generic robust sketch solution (Dodis et al., 2004), where error correction techniques are applied to achieve the fuzzy extractor in the safety sketches and strong random extraction. The problem of non-uniformity of biometric information was addressed by using a hash function. The use of a hash function provides also integrity assurances that transmitted information has not been tampered with by an active adversary. An alternative option is studied in Dodis et al. (2006), where the hash function is replaced by a Message Authentication Code (MAC) algorithm.

Generally speaking, a secure sketch scheme is a procedure that reliably and securely reconstructs a binary string from data of the same source despite the presence of noise (Dodis et al., 2004). Robustness, in the presence of noise, can be achieved by producing helper data, which are re-processed with any fresh input and help, as the name suggests, to retrieve the original data, but only if new input is close enough to the original (Tuyls et al., 2005). Helper data can be hashed if stored in a public database and when the authentication is remote (Boyen et al., 2005; Li et al., 2017; Dodis et al., 2012). A fuzzy extractor scheme further extracts uniform and random binary strings (Dodis et al., 2004); they are designed to be robustly re-usable, that is such that to detect whether an adversary tries to manipulate helper data to get one of its identifiers authenticated (Wen and Liu, 2018; Wen et al., 2019).

A plethora of secure sketch and fuzzy extractor models have emerged, not only for biometric systems (Tuyls et al., 2005; Dodis et al., 2012; Li et al., 2006; Liu et al., 2017; Nandakumar and Jain, 2015b) but also for PUFs (Delvaux et al., 2016; Kang et al., 2014; Mesaritakis et al., 2018). However, there are no once-and-for-all solutions; each source of information has its own peculiarities that require *ad hoc* solutions. However, certain proposals can inspire better than others designs that fit a particular authentication system. CSRs are believed to be optical PUFs (Lenzini et al., 2017), and their peculiarity is that their response can be modeled as circles in a plane. Li et al. (2017) studied secure sketch and fuzzy extractor for systems whose responses are points in a line. We can extend to work for our particular responses, since it requires extending the solution to work in two dimensions. Thus, for our purpose, Li et al. (2017) is the most appropriate proposal we can resort to, as we will prove our generalization will preserve the same security guarantees offered during remote authentication.

Previous work on CSRs analysis. Lenzini et al. (2017) have pioneered extracting features from CSRs by comparing the histograms of images. Arenas et al. (2021, 2022b) proposed two metrics, one based on CSRs image subtraction and the other one based on a comparison of the extracted minutiæ. With varying degrees of confidence, these contributions prove that it is feasible to recognize noisy images of the same CSR and to distinguish them from different actual CSRs and fake images. However, procedures that allow extracting robust information from CSRs and stronger security guarantees (e.g., non-invertibility) are still missing. In biometrics, such schemes are well-studied. Those closely related to this work are Li et al. (2017), Tuyls et al. (2005), Canetti et al. (2021). Among them, Tuyls et al. (2005) proposed to cope with the variability of intra- and inter-subject fingerprint images by using statistical analysis to robust minutiæ, an idea that this work adapts to CSRs.

Like in the case of biometric systems, the primary use case of the CSR technology is the design and development of secure authentication protocols for verifying the authenticity of an object. The first such application was presented by Arenas et al. in Arenas et al. (2022a). The protocol applies a HE scheme to compare noisy inputs in the authentication phase to reference data stored during the enrollment phase. The comparison is performed in the encrypted domain, thus keeping the data private to the entity performing the comparison. As explained in the previous section, HE is a useful tool for designing privacy-preserving protocols. However, its main drawback is performance, since the homomorphic operations required for comparing encrypted data are expensive in terms of computational cost, even though they are usually outsourced. Therefore, alternative options for authenticating objects are desirable and in this paper, we demonstrate such a solution by deploying secure sketches and fuzzy extractor to construct an object authentication protocol.

3. Secure extraction

Our process to address challenges (i)–(ii) (see Section 1) is exemplified in Fig. 2-a. We assume we can take pictures of a CSR pattern, and extract from it identifying features producing arrays $\boldsymbol{b} = (b_1, \dots, b_m)$ of circles (blobs). Each b_i is a triplet: coordinate of b_i's center and radius.[1] How blobs are extracted is not important here, and we refer to Arenas et al. (2021) for details. Instead, what matters is: 1. to identify a fixed-length list of features (*feature embedding*) with high probability invariant to intra-subjects noise at a level of presence/absence of information (Section 3.1); 2. to extract from it *reliable bits* which are also *robust* from intra-subjects noise; 3. to protect the bitstring from an adversary that aims to retrieve $\boldsymbol{b}$, or any data that links to it, from public data (Sections 3.3 and 3.4).

3.1. Feature embedding

This step ensures that the output of processing images returns an array of exactly n features. This does not happen when processing CSR images: the number of blobs identified varies across images of the same CSR and different CSRs. A fixed size data structure of n features is a necessary condition to apply our secure sketch (see Section 3.2).

We use a mesh of n elements – for instance a rectangular grid of n squares Q_i, where $i = 1, \dots, n$ – to quantize the arrays of features $\boldsymbol{b} = (b_1, \dots, b_m)$. The quantization returns an array $\boldsymbol{\omega} = (\omega_1, \dots, \omega_n)$ of exactly n features. Each ω_i is either a circle or undefined, which we indicate as $\perp$. For all the n squares Q_i, let $J_i \subseteq \{1, \dots, m\}$ be the set of indexes of the elements in $\boldsymbol{b}$ whose center is in Q_i. Then, $\boldsymbol{\omega}$ is built as follows:

$$\omega_i = \begin{cases} \bigcup_{j \in J_i} b_j, & \text{if } J_i \neq \emptyset \\ \perp, & \text{otherwise} \end{cases} \tag{1}$$

Here $\cup$ is an associative and commutative operator: $b \cup b'$ is the smallest circle that contains both b and b'; its center is the barycenter of b and b'.

Robust features. Taking pictures of the *same* CSR—let us denote a set of z pictures by $[\texttt{CSR}]_1, \dots, [\texttt{CSR}]_z$— is a noisy process. This means that the various $\boldsymbol{\omega}^1, \dots, \boldsymbol{\omega}^z$, corresponding to the pictures $[\texttt{CSR}]_1, \dots, [\texttt{CSR}]_z$, may differ in the presence or absence of blobs in certain positions. When this happens in a position, we say that the position is *non-robust*. We are interested in identifying robust positions. There are different ways to define a robust position, the most conservative is the following:

Definition 1. Let $\boldsymbol{\omega}^1, \dots, \boldsymbol{\omega}^z$ be the embedded features extracted from z pictures $[\texttt{CSR}]_1, \dots, [\texttt{CSR}]_z$ of the same CSR, where $\boldsymbol{\omega}^j = (\omega_1^j, \dots, \omega_n^j)$, for $j = 1, \dots, z$. We say that a position i is z-*robust* if $\omega_i^j = \perp$, $\forall j = 1, \dots, z$, or $\omega_i^j \neq \perp$, $\forall j = 1, \dots, z$.

3.2. Secure sketches

Our feature embedding process stabilizes the presence or absence of information in a robust position in arrays of features $\boldsymbol{\omega}^1, \dots, \boldsymbol{\omega}^z$ for a specific CSR. At the level of data, that is, about the coordinates and radii of the various $\omega_i^j \in \boldsymbol{\omega}^j$ for all $j = 1, \dots, z$, there may be still noise. When taking pictures, the object carrying the CSR can be in a different position, or the pattern can vary due to slight changes in the angle of illumination or retro illumination from the microscope, etc. To ensure that the same $\boldsymbol{\omega}$ is reconstructed when taking a picture of the same CSR, we propose to use a secure sketch scheme.

A *secure sketch scheme* (Dodis et al., 2004, 2008) is a pair of randomized functions $(\texttt{SS}, \texttt{Rec})$. When combined they implement an error correction technique that works at the level of bits. The function $\texttt{Rec}$ is used to recover exactly the values of an array of features (of reference) from any noisy version of it. This is possible thanks to some auxiliary information created by $\texttt{SS}$ to the features of reference. The reconstruction works under the condition that the original and the noisy values are "close enough". Formally, $\texttt{SS}$ function inputs a vector $\boldsymbol{\omega}$ and outputs a vector $\boldsymbol{s}$, called *sketch*. The function $\texttt{Rec}$, given $\boldsymbol{\omega}'$ —a noisy version of $\boldsymbol{\omega}$— and the sketch $\boldsymbol{s}$ reconstructs $\boldsymbol{\omega}$ if and only if $\texttt{dist}(\boldsymbol{\omega}, \boldsymbol{\omega}') \leq t$, where $\texttt{dist}$ is a distance over the domain of the input values and t a predefined threshold; it fails, i.e., it is undefined, otherwise.

A key definition for this step and for the following secure sketch scheme is that of *number grid*, that in turn depends on the notion of 'number line' proposed in Li et al. (2017):

Definition 2 (*Number Line*)**.** Let $k \in \{2, 4, 6, \dots\}$ an even natural number, and $a \in \mathbb{N}^+$. A *Number Line* L_a, is an interval in $\mathbb{R}$ which includes $\{\dots, -2ka, -ka, 0, +ka, +2ka, \dots\}$. These k-points identify on L_a *intervals* $I_x = (x - \frac{ka}{2}, x + \frac{ka}{2})$. The point x ranges over ka, which is also the central point of the I_x. Each interval has length ka.

Definition 3 (*Number Grid*)**.** A *number grid* G_a is defined as $L_a \times L_a$, where L_a is a Number Line.

Therefore, G_a is a set of open squares $Q_{(x,y)} = I_x \times I_y$ for x and y ranging over the k-points of L_a. Definition 3 can be further generalized to include additional dimensions. An example of a Number Grid is illustrated in Fig. 2-(b). $\overline{Q}_{(x,y)} = (x, y)$ is the midpoint of square $Q_{(x,y)}$. Each square measures $ka \times ka$ pixels and the grid can be aligned to a predefined reference point in the picture. Parameters k and a are chosen so that each square is large enough to contain at least one blob, which

[1] Here, we ignore the information coming from color: it adds three more dimensions in the features spaces, increasing unnecessarily the complexity. Our scheme can be easily extended to work with more dimensions in future work.

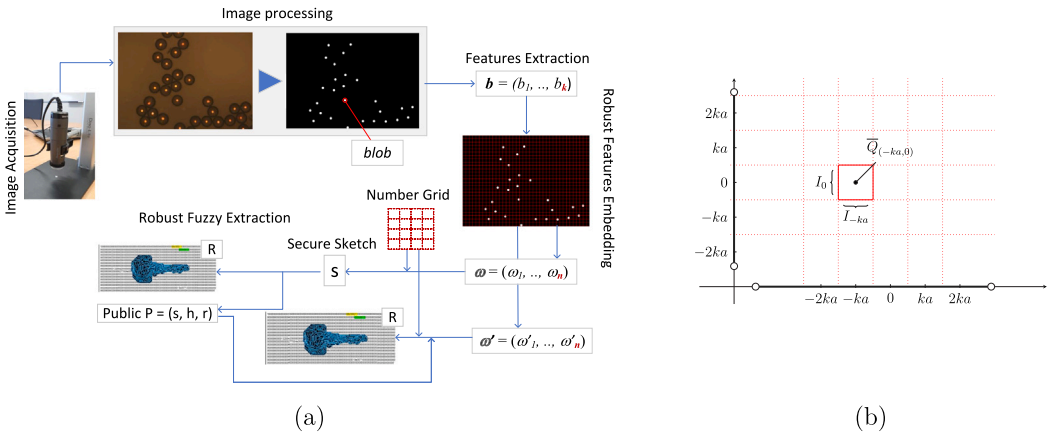


Fig. 2. (a) After the acquisition, CSR images are processed and features extracted as an array of blobs. From there, we extract a fixed size array $\boldsymbol{\omega}$ of reliable features (quantization), we use a secure sketch scheme to generate a bitstring from which, given a new image and the sketch, we can reconstruct the original $\boldsymbol{\omega}$ despite intra-subject noise. Fuzzy extraction is instead used to generate a uniformly random string R that we use in enrollment and authentication. R can be reconstructed using public data s, h, and r and a new set of images. (b) A Number Grid.

in practice means that ka should be larger, in pixels, than the diameter of an average blob. Our proposed secure sketch assumes a metric space (G_a, dist), (see Definition 3), with dist being the Chebyshev distance:

Definition 4 (*Chebyshev Distance*)**.** Let $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$, $\boldsymbol{\omega}' = (\omega'_1, \ldots, \omega'_n)$ be two vectors, with $\omega_i = (x_{\omega_i}, y_{\omega_i}) \in \mathbb{Z}^2$ and $\omega'_i = (x_{\omega'_i}, y_{\omega'_i}) \in \mathbb{Z}^2$. The *Chebyshev distance* between $\boldsymbol{\omega}$ and $\boldsymbol{\omega}'$, is the function $\mathsf{dist} : \mathbb{Z}^{2\times n} \times \mathbb{Z}^{2\times n} \to \mathbb{N}$ such that:

$$\mathsf{dist}(\boldsymbol{\omega}, \boldsymbol{\omega}') = \max_{i=1,\ldots,n} \left\{ \max\{|x_{\omega_i} - x_{\omega'_i}|, |y_{\omega_i} - y_{\omega'_i}|\} \right\}.$$

Our secure sketch scheme is defined by its Setup, and by its functions $(\mathsf{SS}, \mathsf{Rec})$.

Setup. The Setup assumes a grid G_a that has $N \times N$ squares so that it contains all the features in the $\boldsymbol{\omega}$s. Further, following the secure sketch construction of Li et al. (2017), we set the *maximum acceptance Chebyshev threshold* to t which is a value less than $\frac{ka}{2}$.

Sketch construction. The function SS takes as input a robust template $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$ of a CSR image and returns the sketch, a vector $\boldsymbol{s} = (s_1, \ldots, s_n)$. Each element of $\boldsymbol{s}$ is a pair of values $s_i = (x_{s_i}, y_{s_i}) \in \{0,1\}^* \times \{0,1\}^*$ (i.e., x_{s_i} and y_{s_i} are numbers in finite representation form) or the undefined symbol $\perp$. The sketch $\boldsymbol{s}$ is constructed as follows. For each $\omega_i \in \boldsymbol{\omega} \setminus \{\perp\}$, $s_i = (x_{s_i}, y_{s_i})$ is the offset to move ω_i to the closest middle point of the square that contains the center of ω_i. Formally, $x_{\overline{Q}_i} = x_{\omega_i} + x_{s_i}$ and $y_{\overline{Q}_i} = y_{\omega_i} + y_{s_i}$, where $|x_{s_i}|$ and $|y_{s_i}|$ are both $\leq \frac{ka}{2}$. Instead, whenever $\omega_i = \perp$, then $s_i = \perp$. If an ω_i's center lays on the external border of a square, we toss a coin to choose, as a reference, the middle point of one of the neighbor squares. For instance, if the center lay on the horizontal (resp. vertical) border of a square and the coin is 'head' we chose the square above (resp. on the left); instead, if the coin is 'tail' we chose the square below (resp. on the right). If the center is one of the borders of the grid, we consider the grid folded on itself on both dimensions, and we select the square accordingly.

Reconstruction. The function Rec takes as input a retake $\boldsymbol{\omega}' = (\omega'_1, \ldots, \omega'_n)$ of a CSR image and a sketch $\boldsymbol{s} = \mathsf{SS}(\boldsymbol{\omega})$, where $\boldsymbol{s} = (s_1, \ldots, s_n)$. It returns a vector $\boldsymbol{z}$, which is equal to $\boldsymbol{\omega}$, or fails.

For all $\omega'_i \in \boldsymbol{\omega}' \setminus \{\perp\}$, if $s_i \neq \perp$ the function computes $x_{v_i} = x_{\omega'_i} + x_{s_i}$ and $y_{v_i} = y_{\omega'_i} + y_{s_i}$. If for one or both dimensions x and y of v is bigger (resp. smaller) than $N\frac{ka}{2}$ (resp., $-N\frac{ka}{2}$), which means the point falls outside the boundary of G_a, we subtract from (resp. add to) it the value ka. For instance if $x_{v_i} > N\frac{ka}{2}$ and $y_{v_i} < -N\frac{ka}{2}$ then $(x_{v_i}, y_{v_i}) = (x_{v_i} - ka, y_{v_i} + ka)$.

For all such v_i, assuming that Q is the square that contains the center of v_i, the function computes $z_i = (x_{z_i}, y_{z_i})$ or $\perp$ as follows

$$(x_{z_i}, y_{z_i}) = \begin{cases} (x_{\overline{Q}} - x_{s_i}, y_{\overline{Q}} - y_{s_i}), & \text{if } (|x_{\overline{Q}} - x_{v_i}| < t) \wedge (|y_{\overline{Q}} - y_{v_i}| < t) \\ \text{abort}, & \text{otherwise} \end{cases}$$

If, for any i, the reconstruction does not abort, i.e., if $\mathsf{dist}(\boldsymbol{\omega}, \boldsymbol{\omega}') \leq t$, then Rec returns $\boldsymbol{z} = (z_1, \ldots, z_n)$, and it will be exactly $\boldsymbol{\omega}$.[2] The correctness of our secure sketch $(\mathsf{SS}, \mathsf{Rec})$ is described in the following Theorem.

Theorem 1. *Let $\boldsymbol{\omega}$ and $\boldsymbol{\omega}'$ be two arrays of features, t be the maximum acceptable Chebyshev threshold, and $\boldsymbol{s} = \mathsf{SS}(\boldsymbol{\omega})$ be a secure sketch. Then $\mathsf{Rec}(\boldsymbol{\omega}', \boldsymbol{s}) = \boldsymbol{\omega}$ if and only if $\mathsf{dist}(\boldsymbol{\omega}, \boldsymbol{\omega}') \leq t$.*

The proof is an extension into two dimensions of the proof given by Li et al. (2017, Theorem 1, p. 5) (see Appendix).

3.3. Robust secure sketch

Boyen et al. (2005) noticed that when a normal secure sketch scheme $(\mathsf{SS}, \mathsf{Rec})$ is deployed in a cryptographic application, it does not provide any protection mechanism against the deliberate modification of the sketch $\boldsymbol{s} = \mathsf{SS}(\boldsymbol{\omega})$ by an active adversary. They remedy this problem by introducing the notion of *robust secure sketches*, initially introduced by Boyen et al. (2005).

A robust secure sketch scheme is a pair of functions (RobustSS, RobustRec) (see Algorithms 1 and 2). The difference from a normal secure sketch $(\mathsf{SS}, \mathsf{Rec})$ is that in addition to the sketch $\boldsymbol{s}$, the function RobustSS computes and outputs the digest of the vectors $\boldsymbol{\omega}$ and $\boldsymbol{s}$, using a cryptographic hash function $H : \{0,1\}^* \to \{0,1\}^l$, where l is the size of the digest and depends on the chosen hash function. A party that receives $(\boldsymbol{s}, h)$ can verify that the sketch has not been tampered with, by reconstructing $\boldsymbol{\omega}$ using the Rec function and recomputing the digest, which is compared against the received h.

Algorithm 1: RobustSS.

Input: $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$.
Output: $\boldsymbol{s}, h$.
1 $\boldsymbol{s} = (s_1, \ldots, s_n) \leftarrow \mathsf{SS}(\boldsymbol{\omega})$
2 $h \leftarrow H(\boldsymbol{\omega}, \boldsymbol{s})$
3 **return** $(\boldsymbol{s}, h)$

[2] As we are interested only in the reliable positions, the reconstruction aborts if it fails to reconstruct the template with a minimum number of reliable positions.

Algorithm 2: RobustRec.

Input: $\boldsymbol{\omega}' = (\omega'_1, \ldots, \omega'_n), s, h$.
Output: z, or $\perp$.
$z \leftarrow \mathsf{Rec}(\boldsymbol{\omega}', s)$
if $z = \perp$ **then**
  return $\perp$
else
  if $h \neq H(z, s)$ **then**
    return $\perp$
  else
    return z
  end
end

3.4. Fuzzy extractors

Fuzzy extractors were proposed by Dodis et al. in Dodis et al. (2004, 2008) as a means for extracting a (uniformly) random string R from a given input. Their crucial property is that they are error-tolerant, meaning that the string R can be reconstructed from a noisy input as long as it is relatively close to the original input. The extracted string R can be used in cryptographic applications, for instance, as an encryption key for symmetric algorithms or as a seed for generating a pair of secret/public keys for asymmetric algorithms. We highlight that the ability to reconstruct the string R eliminates the need for secure storage and management of cryptographic keys, thus reducing the complexity of the cryptographic application.

A fuzzy extractor operates on top of a secure sketch scheme. It is composed of a Setup phase and a pair of randomized functions $(\mathsf{Gen}, \mathsf{Rep})$, which are described in Algorithms 3–4. For the Setup phase, we assume a robust secure sketch $(\mathsf{RobustSS}, \mathsf{RobustRec})$ over the metric space (G_a, dist), and a *strong randomness extractor* (Nisan and Zuckerman, 1996) defined as $\mathsf{Ext} : \{0,1\}^* \times \{0,1\}^m \rightarrow \{0,1\}^l$. Strong randomness extractors are essentially families of hash functions, used to convert high-entropy inputs into shorter uniformly distributed digests.

Algorithm 3: Gen.

Input: $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$.
Output: (R, P).
$r \xleftarrow{\$} \{0,1\}^m$
$(s, h) \leftarrow \mathsf{RobustSS}(\boldsymbol{\omega})$
$R \leftarrow \mathsf{Ext}(\boldsymbol{\omega}, r)$
$P \leftarrow (s, h, r)$
return (R, P)

Algorithm 4: Rep.

Input: $\boldsymbol{\omega}' = (\omega'_1, \ldots, \omega'_n)$, P.
Output: R.
$z \leftarrow \mathsf{RobustRec}(\boldsymbol{\omega}', s)$
if $z = \perp$ **then**
  return $\perp$
else
  $R \leftarrow \mathsf{Ext}(z, r)$
  return R
end

The function Gen takes as input a robust template $\boldsymbol{\omega}$ of a CSR and uses the function RobustSS to produce a sketch s and the digest h. It proceeds by using the strong extractor Ext to create a secret string R, on input the vector $\boldsymbol{\omega}$ and a random value $r \in \{0,1\}^m$. The output of Gen is the string R and the triple $P = (s, h, r)$. We refer to the latter as the *public helper data*, since all of its components are public values used to reconstruct the secret R. On input a retake $\boldsymbol{\omega}'$ of the CSR and the public helper data $P = (s, h, r)$, the function Rep aims at reproducing the secret value R. It executes the RobustRec algorithm which outputs either a vector z which matches the original vector $\boldsymbol{\omega}$, or it is $\perp$. If z is indeed a vector, Rep proceeds in recomputing the secret $R \leftarrow \mathsf{Ext}(z, r)$.

4. Authentication protocols

The procedures described in Section 3.2 are building blocks to realize two security protocols. The first, *enrollment*, describes how to register a CSR-carrying object csrID to a remote server; the other, *authentication*, defines how to verify whether an object, presumably csrID, is the object csrID previously registered. We assume four entities:

- ***Service Provider* (SP)**: It takes pictures $[\mathtt{CSR}]_i$ of the csrID for enrollment and sends them to a device for image and data processing. The SP is the entity having the right to enroll the CSR images.
- ***User* (U)**: It takes pictures $[\mathtt{CSR}']_i$ of an object csrID and sends them to the device for image and data processing. The *User* queries the server to initiate the authentication process.
- ***Authentication Device* (AD)**: Trusted device that processes images $[\mathtt{CSR}]_i$ by applying the feature embedding, robust secure sketch, and fuzzy extractor to produce the data for enrollment. It does similar steps, plus other checks, for authentication. It communicates with the authentication server initiating the enrollment or the authentication.
- ***Authentication Server* (AS)**: It responds to enrollment and authentication requests. It maintains a database of csrIDs and public data needed for authenticating a csrID.

4.1. Enrollment

The enrollment is carried out by the SP, the AD, and the AS (see Fig. 3). It is initiated by the SP which submits to the AD a set of CSR images $[\mathtt{CSR}]_i$ along with a unique identifier csrID. Upon receiving the information from the SP, the AD executes a series of actions. The first is to extract the robust template $\boldsymbol{\omega}$ from the given set of CSR images $[\mathtt{CSR}]_i$. This requires the steps described in Section 3.1, namely the image processing, features embedding and robust features. In Fig. 3, we combine all three steps in a single function and write $\boldsymbol{\omega} \leftarrow \mathsf{ImgProcess}([\mathtt{CSR}]_i)$.

Next, the AD executes the function Gen of the fuzzy extractor scheme, on input the vector $\boldsymbol{\omega}$. According to Algorithm 3, this function returns a secret value R and the public helper data $P = (s, h, r)$. The secret value R is used to generate a pair of private/public signing keys $(ssk_{\mathrm{AD}}, psk_{\mathrm{AD}}) \leftarrow \mathsf{SignKeyGen}(R)$. The AD signs the message $m = (\mathtt{csrID}, P, psk_{\mathrm{AD}})$ with its secret signing key ssk_{AD}, transmits the csrID, P, psk_{AD} along with the signature σ_{AD} to the AS and discards all processed information. The first action of the AS is to verify the integrity of the transmitted data, by verifying the signature σ_{AD}. If the verification is successful, the AS stores the triple $(\mathtt{csrID}, P, psk_{\mathrm{AD}})$.

4.2. Authentication

The parties involved in the authentication phase are the User, the AD, and the AS (see Fig. 4). It is initiated by the User who submits to the AD a set of fresh images $[\mathtt{CSR}']_i$ along with the identifier csrID. The AD forwards the csrID to the AS requesting the entry corresponding to this identifier, which AS has stored. If an entry exists, the AS forwards the public helper data P to the AD and a freshly generated nonce n_{AS}. Otherwise, the AS returns a message to the AD indicating that the authentication has failed.

Upon receiving P and n_{AS}, the AD extracts the robust template $\boldsymbol{\omega}'$ from $[\mathtt{CSR}']_i$. Then, it executes the Rep function of the fuzzy extractor scheme, with input $\boldsymbol{\omega}'$ and the public helper data P, to reproduce the secret value R. Next, it generates a random nonce n_{AD} and uses the secret value R for recreating the private/public signing key pair $(ssk_{\mathrm{AD}}, psk_{\mathrm{AD}})$. The AD uses the ssk_{AD} to sign the two nonces and the csrID and sends the signature σ_{AD} along with n_{AD} to the AS. The AS verifies the signature on the message $(n_{\mathrm{AD}} \| n_{\mathrm{AS}}, \mathtt{csrID})$, using the public signing key psk_{AD} that was stored in the database during the enrollment phase. Before sending this information to the AD, the AS signs the answer (0/1) and the random values $(n_{\mathrm{AD}} \| n_{\mathrm{AS}})$ with its secret

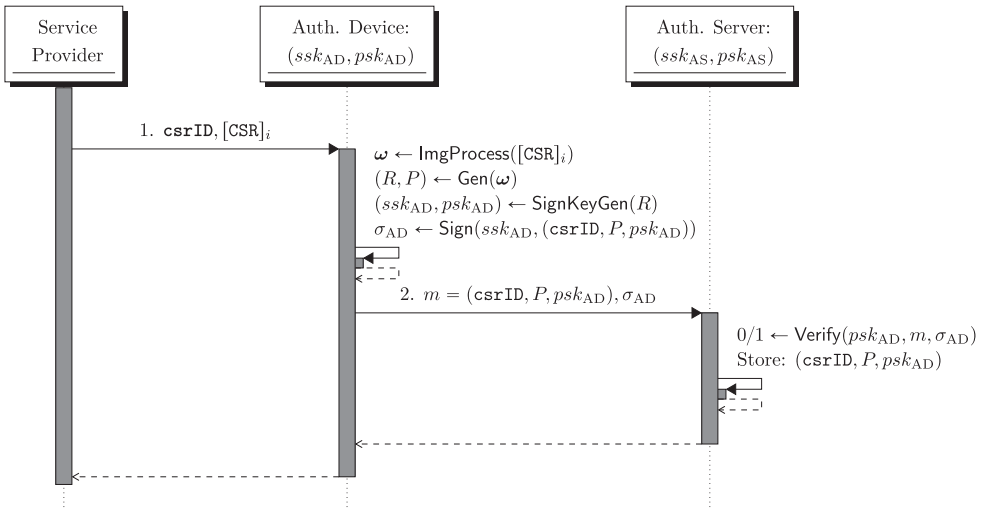


Fig. 3. Message sequence chart of the enrollment phase.

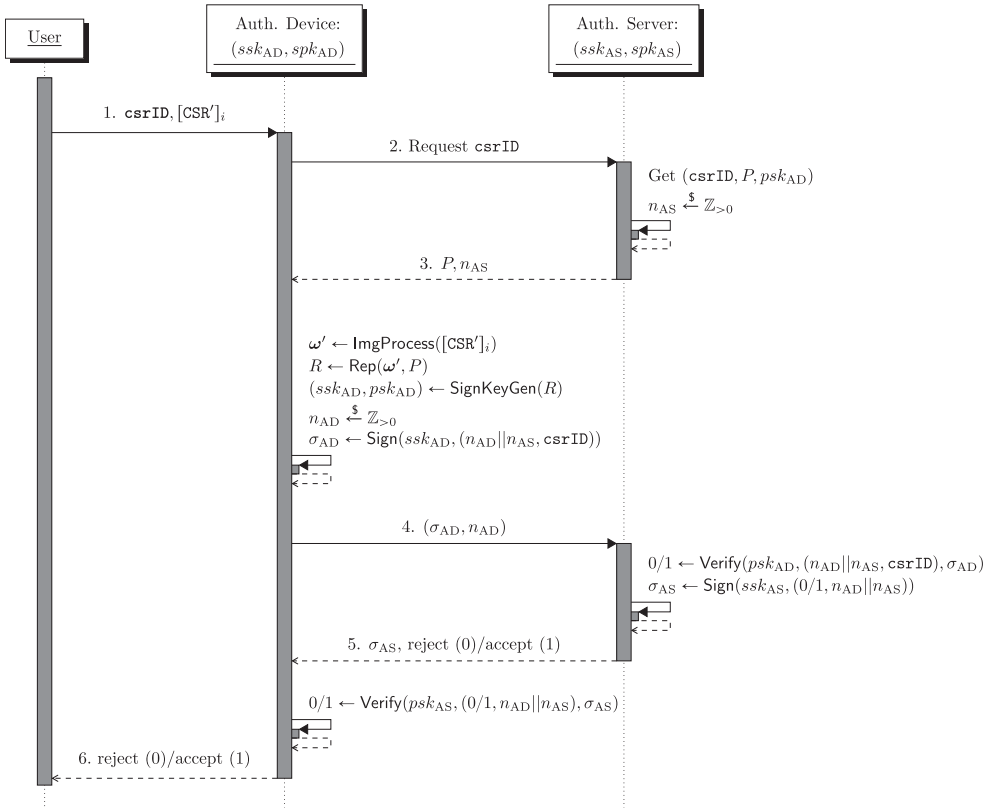


Fig. 4. Message sequence chart of the authentication phase.

signing key (ssk_{AS}). Thus, the AS transmits to the AD the signature σ_{AS} and the answer (0/1). Before sending the answer to the User, the AD verifies the data integrity with the psk_{AD}, and if successful the answer is transmitted. Otherwise, the process is aborted.

5. Security analysis

As we wanted to tailor Li et al.'s (Li et al., 2017) protocol to CSRs, we analyzed its security assumptions and we formally confirmed that their biometric identification protocol meets the secrecy requirements; nevertheless, we detected flaws in its authentication properties. In this section, we have first formally described the security analysis of our authentication scheme, and then, we detail the analysis of Li et al. scheme.

5.1. Security analysis of our scheme

We will show that the cryptographic primitives used in our scheme, the secure sketch and the fuzzy extractor, are secure. Then, we use a symbolic model checker to formally prove the security of our protocol, described in Section 4.

5.1.1. Primitives security

In Li et al. (2017, Theorem 4, p. 9) is shown that the fuzzy extractor is secure if the underlying secure sketch is secure. Thus, we only need

to show that our secure sketch and the corresponding robust secure sketch are secure. The security level of our secure sketch scheme can be specified by evaluating the hardness of inverting the function $s \leftarrow \mathsf{SS}(\boldsymbol{\omega})$, hence recovering a robust template $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$, given the corresponding sketch $s = (s_1, \ldots, s_n)$. We recall two definitions from Dodis et al. (2004, 2008).

Definition 5 (*Average Min Entropy*)**.** Let X be a random variable with a probability distribution. The *min-entropy* of X is:

$$\mathrm{H}_\infty(X) = -\log_2(\max_x\{\Pr(X = x)\}).$$

If X, Y are two discrete random variables, the (conditional) *average min-entropy* of X given Y is:

$$\tilde{\mathrm{H}}_\infty(X|Y) = -\log_2\left(\mathbb{E}_{y\leftarrow Y}\left[\max_{y\in Y}\{\Pr(X = x|Y = y)\}\right]\right).$$

Definition 6. A secure sketch scheme is $(\mathcal{M}, m, \tilde{m}, t)$-*secure*, if for any distribution X over the metric space $\mathcal{M}$ with min-entropy m, the advantage of an adversary in recovering the value of X is at most $2^{-\tilde{m}}$, where $\tilde{m} \leq \tilde{H}_\infty(X|\mathsf{SS}(X))$.

Based on the above definitions, we conclude to the next theorem on the security of our secure sketch scheme.

Theorem 2 (*Security of Secure Sketch*)**.** *The proposed* (SS, Rec) *scheme is a* $(G_a, 2n\log_2(kaN), 2n\log_2 N, t)$*-secure sketch, where* $n = N \times N$ *is the number of squares in the grid* G_a*. The entropy loss is* $2n\log_2(ka)$ *and the storage is* $2n(\lfloor\log_2 ka\rfloor + 1)$*. Both algorithms* SS, Rec *run in polynomial time in* n, k, a, N.

The proof follows similar arguments as those reported in (Li et al., 2017, Theorem 3, p. 8), which we extend to our scheme (see Appendix). The security of the proposed secure sketch depends on the size n of the robust template vector $\boldsymbol{\omega}$ and the number of squares in the grid G_a. As pointed out in Li et al. (2017), the robust secure sketch (RobustSS, RobustRec) is secure as long as the basic (SS, Rec) scheme is secure. The two versions differ only because of the additional computation of the digest $h \leftarrow H(\boldsymbol{\omega}, s)$ in RobustSS function, and in the verification of this digest in RobustRec. Hence, the security of (RobustSS, RobustRec) relies on the security properties of the underlying hash function (one-wayness and collision resistance).

Furthermore, in the context biometric authentication systems the concept of *relative entropy* (Adler et al., 2009) is used as an appropriate quantity and a more realistic way to evaluate the security of a biometric system (see for example Zhu et al. (2023), Takahashi and Murakami (2013), Youmaran and Adler (2012), Al-Assam et al. (2012)). The relative entropy of the biometric information of a user measures the amount of information (in bits) that distinguishes the user from a given population. This is usually approximated using probabilistic models that describe the similarity of user's biometric information with the biometric information in a given dataset and the larger the dataset, the more accurate the approximation.

As the CSR technology resembles the biometric technology, such a measurement of the entropy of CSRs would certainly fit in our security analysis. However, the CSR technology is currently at a less matured stage and our CSR dataset is rather small to conduct such an analysis and extract useful conclusions. We leave such an in-depth analysis of the relative entropy of CSRs for future work, where a richer dataset is expected to be available, as the team plans to further invest in this technology and its integration in the context of anti-counterfeiting protocols. Nevertheless, in Section 6, we compare the entropy of our proposed secure sketch with the proposal in Li et al. (2017), in terms of the min-entropy, average min-entropy and entropy loss.

Table 1
Equational theory for modeling our protocols.

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

5.1.2. Protocols security

We analyze the security of our protocols using ProVerif, a model-checking software (Blanchet, 2001). Cryptographic primitives can be modeled as functions in terms of equational theories. The equational theories used in our protocol are listed in Table 1.

ProVerif follows a Dolev–Yao threat model (Dolev and Yao, 1983) which considers the following assumptions: (i) all cryptographic primitives are perfect and (ii) an attacker can eavesdrop, block, reply, or manipulate data on public communication channels. Security requirements that we aim to satisfy in our protocol can be categorized into two properties, secrecy and authentication. To prove secrecy properties in the protocol, we use the reachability queries on sensitive parameters; the CSR template $[\mathtt{CSR}]_i$, the robust template $\boldsymbol{\omega}$, the fuzzy extractor secret string R, and the secret singing key ssk. In ProVerif, authentication properties are modeled using correspondence assertions. First, we define all *events* used to build the correspondence assertions. In the applied π-calculus, an event is an internal message, which together with its arbitrary arguments acts as a flag to capture the state of a process at a precise location within a protocol trace. Events have no effect on the protocol behavior and are only used to reason about the reachability of a state. The events in our CSR authentication protocol and the situations in which they are emitted are listed as follows:

- enrolRequested(csrID,P,psk): when AD sends the enrollment request (csrID,P,psk) to AS.
- enrolVerified(csrID,P,psk): when AS verifies the enrollment request sent by an authentic AD for a CSR object with the identity of csrID, public helper data of P and public signing key of psk.
- authRequested(csrID,P,$\mathtt{n}_{\mathtt{AD}}$,$\mathtt{n}_{\mathtt{AS}}$): when AD sends the authentication request to AS with ($\mathtt{n}_{\mathtt{AD}}$,$\mathtt{n}_{\mathtt{AS}}$) as protocol nonces for a CSR object with the identity of csrID and public helper data of P.
- authVerified(csrID,P,$\mathtt{n}_{\mathtt{AD}}$,$\mathtt{n}_{\mathtt{AS}}$): when AS verifies the authentication request sent by an authentic AD with ($\mathtt{n}_{\mathtt{AD}}$,$\mathtt{n}_{\mathtt{AS}}$) as protocol nonces for a CSR object with the identity of csrID and public helper data of P.
- resultSent(res,$\mathtt{n}_{\mathtt{AD}}$,$\mathtt{n}_{\mathtt{AS}}$): when AS sends res as the result for the protocol with ($\mathtt{n}_{\mathtt{AD}}$,$\mathtt{n}_{\mathtt{AS}}$) nonces.
- resultVerified(res,$\mathtt{n}_{\mathtt{AD}}$,$\mathtt{n}_{\mathtt{AS}}$): when AD verifies res as the result sent by AS for the protocol with ($\mathtt{n}_{\mathtt{AD}}$,$\mathtt{n}_{\mathtt{AS}}$) nonces.

We define three authentication properties for our CSR authentication protocol. These properties are defined in terms of correspondence over the above events.

Definition 7 (*Enrollment Authentication*)**.** The protocol ensures *Enrollment Authentication* if for each occurrence of the event enrolVerified, there is a distinct earlier occurrence of the event enrolRequested.[3]

This property assures that if the AS stores a record in the DB, then this record must have been enrolled by an authentic AD.

$$inj\text{-}\mathsf{enrolVerified}(\mathtt{csrID}, \mathtt{P}, \mathtt{psk}) \rightsquigarrow inj\text{-}\mathsf{enrolRequested}(\mathtt{csrID}, \mathtt{P}, \mathtt{psk})$$

[3] This in Proverif is expressed as an injective correspondence relation, which captures the one-to-one relationship. Injective events are prefixed with inj.

Definition 8 (*AD Authentication*)**.** The protocol ensures *AD Authentication* if for each occurrence of the event authVerified there is a distinct earlier occurrence of the event authRequested.

Holding AD Authentication means that if the AS successfully verifies an authentication request, then the authentication request has been made by an authentic AD.

$inj\text{-}\mathsf{authVerified}(\mathtt{csrID}, \mathtt{P}, \mathtt{n}_{\mathrm{AD}}, \mathtt{n}_{\mathrm{AS}}) \rightsquigarrow inj\text{-}\mathsf{authRequested}(\mathtt{csrID}, \mathtt{P}, \mathtt{n}_{\mathrm{AD}}, \mathtt{n}_{\mathrm{AS}})$

Definition 9 (*CSR Authenticity*)**.** The protocol ensures *CSR Authenticity* if for each occurrence of the event authVerified there is a distinct earlier occurrence of the event enrolRequested. In other words, CSR Authenticity ensures that if the AS verifies a CSR object, the request for enrollment of this object must have been submitted previously with the same csrID and P.

$inj\text{-}\mathsf{authVerified}(\mathtt{csrID}, \mathtt{P}, \mathtt{n}_{\mathrm{AD}}, \mathtt{n}_{\mathrm{AS}}) \rightsquigarrow inj\text{-}\mathsf{enrolRequested}(\mathtt{csrID}, \mathtt{P}, \mathtt{psk})$

Definition 10 (*Result Authenticity*)**.** The protocol ensures *Result Authenticity* if for each occurrence of the event resultVerified there is a distinct earlier occurrence of the event resultSent. Satisfying Result Authenticity implies that the result received by AD is originally generated by AS.

$inj\text{-}\mathsf{resultVerified}(\mathtt{res}, \mathtt{n}_{\mathrm{AD}}, \mathtt{n}_{\mathrm{AS}}) \rightsquigarrow inj\text{-}\mathsf{resultSent}(\mathtt{res}, \mathtt{n}_{\mathrm{AD}}, \mathtt{n}_{\mathrm{AS}})$

Threat model and security assumptions. In our CSR authentication protocol, we model the attacker as a standard Dolev–Yao adversary (Dolev and Yao, 1983), with additional capabilities that we define below. The following assumptions were made during our protocol analysis:

- **(i)** The attacker has no knowledge of the communication channels between the SP and the AD, or between the User and the AD. These channels are private and accessible only to legitimate parties. However, the communication channel between the AD and the AS is public, allowing the attacker full control over it.
- **(ii)** all entities in the protocol are assumed to be honest. However, the attacker is permitted to participate as a user in the protocol and make any number of arbitrary authentication requests.

Attacker goals. As we earlier described in Section 4, The primary objective of our protocol is to enable legitimate service providers to enroll an object with a trusted server and later allow users to verify the object's authenticity through an authentication protocol. Given these objectives and the attacker's capabilities described above, the attacker's potential goals can be listed as follows:

- Enrolling a fake object in the enrollment phase. In this case, the attacker attempts to enroll an object different from the one intended by the SP.
- Compromising the secrecy of the CSR object. The attacker can achieve this goal by obtaining any of the following: the CSR template $[\mathtt{CSR}]_i$, the robust template ω, or the secret string R.
- Authenticating a fake object during the authentication phase. In this scenario, the attacker attempts to successfully authenticate an object other than the one intended by the user.
- Manipulating the authentication result. The attacker's goal here is to alter the authentication result sent from the AS to the AD.

5.1.3. Heuristic security analysis

Our proposed protocol is primarily designed for object authentication. As a result, the attack vectors are fundamentally different from those of biometric authentication schemes intended for key agreement (Wang et al., 2021; Jiang et al., 2020). Table 2 presents the results of the ProVerif analysis performed under the threat model described in Section 5.1.2. These results confirm that our authentication protocol meets all the secrecy and authentication requirements.[4] We further explain how our protocol mitigates the potential attacks described in Section 5.1.2.

Security against fake object enrollment. During the enrollment phase, the attacker may attempt to enroll a counterfeit object on the server. However, since the request from an honest AD to the AS is signed by the AD's private key, the AS verifies not only the authenticity of the sender, but also the integrity of the transmitted data by checking the accompanying signature. This guarantees that enrolling a fake object in the enrollment phase is impossible. ProVerif analysis confirms that our protocol is resistant to this attack by proving that *Enrollment Authentication* is maintained.

Secrecy of sensitive parameters. Since the AD device communicates with users and service providers through a private channel, the CSR template $[\mathtt{CSR}]_i$ remains confidential from the attacker. Additionally, the robust template ω, calculated by a trusted AD, is unknown to the attacker. As demonstrated in Section 5.1.1, it is impossible to recover the secret string R using only the helper data P. Therefore, the attacker, given the capabilities outlined in Section 5.1.2 does not gain knowledge of the sensitive elements $[\mathtt{CSR}]_i$, ω, and R. Moreover, since the AD does not store any information on the device, there is no risk of secret memory leakage during or after protocol execution.

Security against fake object authentication. We assert that the authentication phase of our protocol ensures *Mutual Authentication* between the AD and the AS, making it impossible for an attacker to authenticate a counterfeit object. Since message 4, sent from the AD to the AS, includes the AD's signature, the AS verifies both the authenticity of the sender and the integrity of the message. Conversely, the AD only accepts authentication results that are signed by the AS, ensuring that the result is genuinely generated by the server and has not been tampered with by an attacker. Consequently, the AD and the AS successfully authenticate each other, and the protocol provides protection against both *Fake Object Authentication* and *Manipulated Result*. These security guarantees are also formally verified by ProVerif as *AD Authentication*, *CSR Authenticity*, and *Result Authenticity*.

Security against replay attack. Both the enrollment and authentication phases of our protocol use fresh nonces, ensuring that each session occurs only once for the corresponding nonces. This mechanism prevents an attacker from reusing messages from previous protocol executions.

5.2. Security analysis of Li et al.'s scheme

In this section, we formally analyze the identification protocol proposed by Li et al. in Li et al. (2017). We show that they fulfill the protocol secrecy properties; however, despite the authors' claim, their scheme cannot guarantee a must-have authentication requirement.

5.2.1. Protocol description

The protocol has the same structure and goal as our CSR authentication scheme (described in Section 4), except that we used CSRs and Li et al. used biometric data. Two *Enrollment* and *Authentication* phases are as follows:

Enrollment phase. User presents its identity ID and biometric Bio to the AD via a private channel. Then, the AD runs the Gen function of a secure fuzzy extractor on Bio input and generates (R, P) as a set of outputs. The helper data P consists of a secure sketch element s and a random string r. Moreover, the AD computes $\mathsf{SignKeyGen}(R) = (ssk, psk)$ as a pair of signing keys for the biometric Bio and sends (ID, psk, P) to the AS through a public channel. Upon receiving (ID, psk, P), the AS stores them in its database.

[4] Dataset and source code are available upon request to the authors.

Authentication phase. User sends its biometric Bio to the AD via a private channel. After receiving Bio, the AD executes the secure sketch function secureSketch on Bio and obtains s' as the output. Then, the AD sends s' to the AS through a public channel. When the AS receives s', it searches a public helper data P in the database records such that $s \approx s'$ where $P = (s, r)$. Now the AS generates a nonce n_{AS} and sends (P, n_{AS}) to the AD. Upon receiving (P, n_{AS}), the AD executes the Rep function of the fuzzy extractor on (Bio, P) inputs to reproduce the signing key string ssk. Afterwards, the AD chooses a nonce n_{AD} and sends $\text{Sign}(ssk, (n_{\text{AS}}, n_{\text{AD}}))$ to the AS. The AS verifies the received signature with the psk key which corresponds to P.

5.2.2. Protocol security

The equational theories defined to analyze the protocol in ProVerif are the same as Section 5.1.2. It is worthwhile mentioning that in this protocol, the input for the Rep function is a biometric Bio while in our protocol it is a robust template ω. The security assumptions made by the authors are: **(i)** the channel between the AD and the AS is public and hence is fully controllable by an adversary. Meaning that an adversary can modify, block, or delete any message transmitted on this channel; **(ii)** an adversary is allowed to have access to public helper data, stored in the database of the AS; **(iii)** all entities are honest.

The security requirements we are going to verify in this protocol are three secrecy properties which are Bio, ssk and R secrecy, and three authentication properties *Enrollment Authentication*, *AD Authentication* and *Biometric Authenticity*. *Bio Authenticity* is the same as Definition 9 while instead of a CSR, we are reasoning about biometric data. Since the events in these three correspondence assertions have different inputs from those defined in Section 5.1.2, we express the assertions with new modified events.

- Enrollment Authentication holds if:
  $inj\text{-}\mathsf{enrolVerified}(\mathtt{ID}, \mathtt{psk}, \mathtt{P}) \rightsquigarrow inj\text{-}\mathsf{enrolRequested}(\mathtt{ID}, \mathtt{psk}, \mathtt{P})$
- AD Authentication holds if:
  $inj\text{-}\mathsf{authVerified}(\mathtt{ID}, \mathtt{P}, \mathtt{n_{AS}}, \mathtt{n_{AD}}) \rightsquigarrow inj\text{-}\mathsf{authRequested}(\mathtt{P}, \mathtt{n_{AS}}, \mathtt{n_{AD}})$
- Biometric Authenticity holds if:
  $inj\text{-}\mathsf{authVerified}(\mathtt{ID}, \mathtt{P}, \mathtt{n_{AS}}, \mathtt{n_{AD}}) \rightsquigarrow inj\text{-}\mathsf{enrolRequested}(\mathtt{ID}, \mathtt{psk}, \mathtt{P})$

ProVerif analyses show that three secrecy requirements hold in the protocol. However, the protocol satisfies no authentication property, as shown in the last row of Table 2. In the original paper, the authors did not formally prove the security of their protocol and informally claimed that it is impossible for an attacker to be identified as a legitimate user without knowing its biometrics. We hereby show an attack trace to refute this claim that clarifies why *Biometric Authenticity* does not hold. Let us assume that the AD sends an enrollment request for the user with (ID, psk, P) parameters to the AS via a public channel. The attacker blocks this message and sends to the AS the modified version of it as (ID, psk_A, P_A) where $\text{Gen}(Bio_A) = (R_A, P_A)$, $\text{SignKeyGen}(R_A) = (ssk_A, psk_A)$ and Bio_A is attacker's biometric. Now the attacker makes a regular authentication request with its biometric Bio_A. The AD takes Bio_A and sends $s'_A = \text{secureSketch}(Bio_A)$ to the AS. Upon receiving s'_A, the AS retrieves the corresponding record in the database which is (ID, psk_A, P_A), and responds to the AD with (P_A, n_{AS}). The AD executes $\text{Rep}(Bio_A, P_A) = ssk_A$ and sends $\text{Sign}(ssk, (n_{\text{AS}}, n_{\text{AD}}))$ to the AS. The AS verifies the received signature with psk_A. This signature verifying key corresponds to ssk_A and hence, the signature verification checks it as true. The whole process means that the attacker can send an authentication request with its biometric Bio_A to the AD and the AS successfully verifies this session for the identity ID which belongs to the victim user. Therefore, the attacker without the knowledge of Bio has impersonated the user with the identity of ID.

Table 2
ProVerif analysis results of our scheme and Li et al. scheme.

	Secrecy				Authentication			
Our scheme	[CSR] ✓	ω ✓	ssk ✓	R ✓	Enr. ✓	AD ✓	CSR ✓	Res ✓
Li et al. (2017)	Bio ✓	– –	ssk ✓	R ✓	Enr. ✗	AD ✗	Bio ✗	– –

6. Proof-of-concept and experiments

We implemented in `Python 3.9.7` the processes shown in Fig. 2-a on a macOS Unix machine.[4] We used a `MacBook Pro` (chip Apple M1), 8-core CPU, and 16 GB RAM.

We implemented the scheme (RobustSS, RobustRec) described in Section 3.3, and the fuzzy extractor (Gen, Rep) described in Section 3.4. We additionally implemented the image processing functions described in Arenas et al. (2021), and the authentication protocol presented in Section 4. As we do not consider quantum-resistance aspects in our analysis, we opt for a simple approach using the hash function SHA-256[5] in the roust secure sketch functions and as the strong extractor Ext in the fuzzy extractor scheme, following the configuration of Boyen et al. (2005). As a digital signature scheme, we chose the ECDSA scheme, instantiated with the elliptic curve NIST P–256.

6.1. Dataset

We analyzed 55 `csrIDs` and, from each one, we acquired a reference CSR image. 44 of them were taken with a professional polarized microscope, with illumination perpendicular to the sample. The other CSR images were taken using a USB `Dino-Lite` digital microscope with flexible LED control and perpendicular illumination to the sample. The images acquired with different microscopes clearly show different CSR images. For instance, Fig. 1-*a* shows well-defined and colored CSRs with almost no external noise. In contrast, Fig. 1-*b* exhibits the typical CSRs features but with additional white circles around the CSRs' cluster due to the reflection of the LED on the sample surface. Therefore, a reliable process must be implemented to extract the information from images exposed to different lighting and environmental conditions and/or acquired with different readout devices.

We generated a set of images by artificially injecting similarity and Gaussian noise in each CSR reference image. The former simulates the noise coming from external conditions, such as rotation, illumination changes, lack of focus, etc. Deledalle et al. (2012). In contrast, Gaussian noise simulates the intrinsic noise of the device due to some failure of the sensors to capture details of the object. As for similarity noise, we introduced the blurring effect (lack of focus), which changes the pixel intensity. Larger pixel windows will blur the image more than smaller ones (Gedraite and Hadad, 2011; Peng et al., 2016). We introduced values ranging from (1×1) up to (3×3) pixel window. For Gaussian noise, we introduced values with a standard deviation ranging from 0.0 to 0.3. These values have been chosen considering realistic conditions by following the methodology described in Arenas et al. (2022b). We implemented enrollment and authentication following two designs: we assumed that enrollment is executed in one attempt, from which seven noisy images were generated. And, for the authentication phase, 20 attempts were considered at different time intervals, while seven images were also acquired from each attempt. The generation of seven images per attempt experimentally ensures that we succeed in extracting robust information.

[5] SHA-256 is currently considered acceptable for hash function applications (Barker and Roginsky, 2019).

Table 3
Parameter selection and secure sketch initialization.

Scheme		N	k	Num. of blobs	Min-entropy[a]	Average[b] min-entropy	Entropy[c] loss	Storage[d]
Li et al. (2017)		5 000	4	–	88 048-bits	44 829-bits	43 219-bits	45 000-bits
Our work	min	15	8	5	3 108-bits	1 758-bits	1 350-bits	2 056-bits
	max	70	24	201	104 999-bits	60 066-bits	44 932-bits	49 256-bits

[a] Min-entropy is computed as $m = n\log_2(kaN)$ for Li et al. (2017) and $m = 2n\log_2(kaN)$ in our case.
[b] Average min-entropy is computed as $\tilde{m} = n\log_2 N$ for Li et al. (2017) and $\tilde{m} = 2n\log_2 N$ for our scheme.
[c] The entropy loss is computed a $m - \tilde{m} = n\log_2(ka)$ for Li et al. (2017) and $m - \tilde{m} = 2n\log_2(ka)$ in our scheme.
[d] The storage requirement is $n(\lfloor\log_2 ka\rfloor + 1) + \log_2 h$ for Li et al. (2017) and $2n(\lfloor\log_2 ka\rfloor + 1) + \log_2 h$ for our scheme.

6.2. Experimental set up and results

As previously mentioned, our dataset is composed of 55 CSR images, each one containing a distinct blob density and different blob diameters. Each image also had a variable size as they were captured with two different read-out devices and also needed a particular magnification setup. Since the images were different, we adjusted the grid G_a accordingly. More precisely, k was chosen depending on the average diameter of the blobs contained in each image, and N in such a way that G_a was large enough to cover the image. We set $a = 1$ to maximize the average min-entropy.

Table 3 (columns 2–4) reports the smallest and the largest parameters in our dataset. In the first case, an image CSR containing 5 blobs with average diameter $k = 8$ is embedded in a grid of 225 squares, while in the second case, 201 blobs with average diameter $k = 24$ were embedded in a grid of 4900 squares. In columns 5–7, we present the lower and upper bounds for the security parameters of the robust secure sketch (RobustSS, RobustRec). Furthermore, we also compare the entropy of our proposed secure sketch scheme, with Li et al.'s biometric secure sketch, with respect to the min-entropy, average min-entropy and entropy loss, as well as the storage requirements.

The min-entropy m for the CSR images varies from 3108-to 104999-bits. The crucial security parameter in a secure sketch is the average min-entropy $\tilde{m}$, which measures the entropy of the robust template $\boldsymbol{\omega}$, when the sketch $\boldsymbol{s}$ is given. For our CSR images, $\tilde{m}$ varies from 1758-to 60066-bits, resulting in an entropy loss $(m - \tilde{m})$ from 1350-to 44932-bits. The last column refers to the maximum space needed for storing the output of RobustSS, namely the sketch vector $\boldsymbol{s} = (s_1, \ldots, s_n)$ and the digest $h = H(\boldsymbol{\omega}, \boldsymbol{s})$, where in our case the latter is 256-bits. The storage requirement for the CSR images that we considered in our experiments ranges from 2056-to 49256-bits.

6.2.1. Computation time

We measured the execution time for the three core operations in the enrollment and authentication phases: image processing (composed of the minutiæ extraction, feature embedding, and robust features); secure sketch; and fuzzy extractor. The experimental results related to the CSR image processing have been comprehensively detailed in a separate publication (Arenas et al., 2021). Fig. 5 reports the time required for each operation, including the total time for enrollment and authentication. It also distinguishes between two cases, corresponding to two different ways of considering what a robust feature is. Recall that during enrollment, we took 7 pictures for the same CSR, in which case we had $\boldsymbol{\omega}^1, \ldots, \boldsymbol{\omega}^7$, from which we identified 7 arrays of robust features. Then we distinguished two cases for identifying the robust features:

Case1 : robust features in those indexes where *all* $\omega_i^1, \ldots, \omega_i^7$ are either $\perp$ or not $\perp$;
Case2 : robust features in those indexes where *the majority* of $\omega_i^1, \ldots, \omega_i^7$ which are either $\perp$ or not $\perp$.

The secure sketch and fuzzy extractor timings in enrollment refer to the functions RobustSS and Gen respectively, while in the authentication phase they refer to the functions RobustRec and Rep. In the secure sketch reconstruction, the maximum acceptable Chebyshev distance was set to $\frac{ka}{2}$.

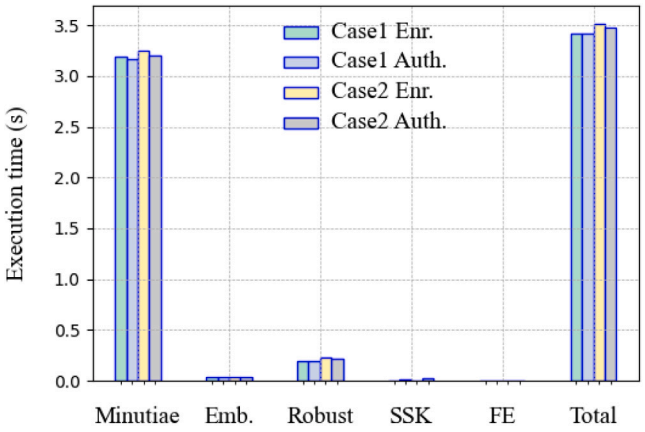


Fig. 5. Average performance (in seconds) of the core operations for both enrollment and authentication phases.

Fig. 5 suggests that the execution time for *Case1* and *Case2* is quite alike. It is evident that the minutiæ detection is the most computationally intensive part in the enrollment and authentication processes. We implemented the algorithm described in Arenas et al. (2021, 2022b); however, we believe that the performance of this procedure can be improved. We observed different performances depending on the 'blob_log' parameters,[6] i.e., and the parameters that constrain the detection of the minimum and maximum diameters of the blobs. We manually tailored these parameters for each CSR image; however, they can be further optimized. The execution time is largely affected by the image's size and the blob density. On average, the minutiæ detection per CSR image consumes 93% of the total execution time in both the enrollment and authentication phases. In contrast, the secure sketch operations take an average runtime of 16 ms and the fuzzy extractor functions take approximately 0.86 ms.

Although our authentication system provides a high level of security, we observed that achieving high performance is challenging because of the implementation overhead introduced mainly by two factors: image processing and the error correction process responsible for stabilizing a noisy response.

6.2.2. Accuracy and robustness

We compared each reference CSR image against twenty attempts from the same CSR (*intra-subject* comparison); and further compared eight randomly selected CSRs against 75 non-correlated CSR images (*inter-subjects* comparison). In total, we made 4400 pairwise comparisons for both intra- and inter-subject experiments, also considering *Case1* and *Case2*.

[6] https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_blob.html

Table 4
Confusion matrix for *Case1* and *Case2*.

		True Clases			
		Accepted		Rejected	
		Case 1	Case 2	Case 1	Case 2
Rslt	Accepted	1064	1086	0	0
	Rejected	36	14	1100	1100

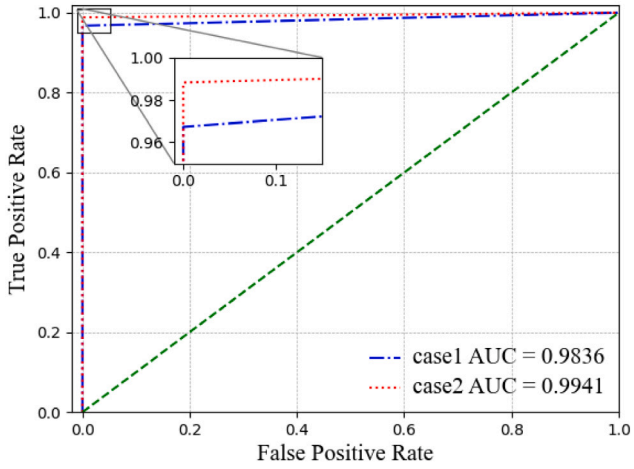


Fig. 6. Receiver operating characteristic ROC curve.

These intra- and inter-subjects comparisons allow us to calculate the confusion matrix and the receiver operating characteristics (ROC) graph. For the former, Table 4 shows an authentication rate of 96.73% (1064/1100) for *Case1* compared to 98.72% (1086/1100) for *Case2*. *Case2* performs slightly better than *Case1*. Regarding false negatives, we noticed that *Case1* rejected 36 responses versus 14 responses for *Case2*. All *inter-subject* comparisons were correctly classified as true-negatives. This result demonstrates that the robust secure sketch scheme rejects `csrIDs` that differ from the enrolled data.

We compared the relative trade-off between true positives (robustness) and false positives (security) for *Case1* and *Case2* as a ROC curve; see Fig. 6. We also computed the area under the ROC curve (AUC), which has a single scalar value (between 0 and 1) that represents the expected performance. We observed that the AUC for *Case1* and *Case2* behave similarly; 0.9836 for the first case against 0.9941 for the latter. We also observed a step curve behavior in Fig. 6, due to the binary nature of our system, i.e., 1 if the signature is validated, 0 otherwise. The results show that *Case2* provides a slight performance gain compared to the stricter assumptions on identifying the robust positions in *Case1*.

7. Conclusions and future work

We have studied the applicability of microscopic droplets of cholesteric liquid crystals as a source of identifying information and their use in object authentication. Used as a coating for objects, they reflect light and respond with colorful patterns whose features can be extracted and represented as a matrix of colored circles. From it, we can produce secure yet reproducible identification tokens. This work shows how to securely merge these unique identifiers with cryptographic primitives to design a remote authentication protocol for objects.

We propose a robust secure sketch scheme and a robust feature extraction scheme to process CSRs, as well as to generate stable and robust bitstrings in the presence of noise. We analytically prove their security. In addition, based on these building blocks, we design a remote and secure authentication protocol composed of two phases: enrollment and authentication.

We verify that the protocols satisfy secrecy and authenticity properties despite a Dolev–Yao adversary who can also manipulate physical objects. We use ProVerif as a model-checker for the formal analysis.

Although, at least in theory, several other schemes could have been used here, not all of them work with the particular optical responses that CSRs return. The challenge that we faced was finding not only a theoretical solution, but also a solution that has, so to speak, a technological level such that it is working as a proof-of-concept. Thus, in designing our scheme secure sketch and fuzzy extractor and the authentication protocols, our aim was to be theoretically clear with a high level of security and to comply with the physical requirements resulting from the optical properties of CSRs and its behavior as a physical unclonable function.

Our schemes are also efficient, as we prove in our proof-of-concept application: analyzed for reliability, quality of authentication, time complexity, and performance, our implementation's computational burden is due to the modules that implement image processing and feature extraction. The time to compute secure sketch and fuzzy extraction is negligible.

Limitations

Our solution to authenticate objects is quite promising, as we addressed the problem of end-to-end secure design, including the physical object without abstraction from reality. In fact, we were able to implement a proof-of-concept and test it. Still, the problem *per se* is not solved once for all, and it would be naïve to think otherwise. There are open problems and challenges coming from handling a physical object and the nature of the noise that exists when taking pictures of a CSR's response. The "objects" we used in our experimental setting have a flat surface, and the pictures were taken using the same camera. In our analysis, we considered a Gaussian type of noise, such as those due to different cameras, but we never tested the solution in a real, out-of-the-lab context. We speculate that our secure sketch and fuzzy extractor schemes should be robust to work in general because, whatever setting, even whatever different technology is used to produce CSRs, their responses are circles in a plane, but we have not set up such an experiment.

Theoretically, our scheme and our protocols have been proven secure, and at least unless we change the adversarial model, that result should be robust. Our code and proofs are available for the community to verify and replicate our results (we may need to ship the material for a full replication, but researchers are invited to contact us for that). However, we assumed that our device is trusted, and that is a strong assumption, especially if the device is a general-purpose phone used not only to authenticate goods. A fully fledged secure authentication would require a secure implementation and thus a change in abstraction from protocol to software security.

Future work

There are several future steps to advance the research we presented here. One is about the precondition for our secure sketch scheme, that is our G_a. Different instances of the grid, with different values for a can influence the number of points in each of the grid's squares and lead to different authentication levels and performances. Another is to remove the bottleneck due to feature extraction since, as pointed out earlier, we believe that the image processing phase and the feature extraction processes can be further optimized. We also believe that it is worth extending the use of CSR in other application domains beyond remote authentication, for instance, to create physical tokens that allow the generation of cryptographic keys for different cryptographic functionalities, e.g., for encryption purposes (either symmetric or public-key). Finally, a comprehensive analysis of the entropy of CSRs will be conducted once a larger dataset is available, as well as a comparison with other authentication factors, such as PINs, passwords, and biometric information, in terms of entropy.

CRediT authorship contribution statement

Mónica P. Arenas: Investigation, Methodology, Software, Writing – original draft, Writing – review & editing. **Georgios Fotiadis:** Investigation, Methodology, Software, Writing – original draft, Writing – review & editing. **Gabriele Lenzini:** Funding acquisition, Investigation, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Mohammadamin Rakeei:** Formal analysis, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the GitLab link on the manuscript.

Acknowledgments

The authors would like to acknowledge: (i) the ESMP group for providing the CSR images and (ii) the financial support from the Luxembourg National Research Fund (FNR) on the projects: (a) No more Fakes "NoFakes" (PoC20/15299666/ NOFAKES-PoC) (b) Secure, Quantum-Safe, Practical Voting Technologies "EquiVox" (C19/IS/13643617/ EquiVox/Ryan), and (c) the INTER project Secure and Verifiable Electronic Testing and Assessment Systems –SEVERITAS (INTER /ANR/ 20/ 14926102 ANR-20- CE39-009-03).

Appendix. Proof of Theorems

Theorem 1. Let $\boldsymbol{\omega} = (\omega_1, \dots, \omega_n)$ be a robust template corresponding to the original CSR image, where each $\omega_i = (x_{\omega_i}, y_{\omega_i}) \in \mathbb{Z}^2$ is point in the number grid G_a. Let $\boldsymbol{\omega}' = (\omega'_1, \dots, \omega'_n)$ be a robust template corresponding to a retake of the original CSR image, where $\omega'_i = (x_{\omega'_i}, y_{\omega'_i}) \in \mathbb{Z}^2$ is point in the number grid G_a. We assume that $\boldsymbol{s} = (s_1, \dots, s_n)$ is the output of the function SS, such that $s_i = (x_{s_i}, y_{s_i}) \in \mathbb{Z}^2$, with $|x_{s_i}|, |y_{s_i}| \leq \frac{ka}{2}$. And, let $t < \frac{ka}{2}$ be the maximum acceptable Chebysev threshold. There are two cases to consider for $\mathsf{dist}(\boldsymbol{\omega}, \boldsymbol{\omega}')$.

First, we assume that $\mathsf{dist}(\boldsymbol{\omega}, \boldsymbol{\omega}') \leq t$. By Definition 4, we have $|x_{\omega_i} - x_{\omega'_i}| \leq t$ and $|y_{\omega_i} - y_{\omega'_i}| \leq t$, for all $i = 1, \dots, n$. For the x-coordinates, the secure sketch function SS implies that $x_{\overline{Q}_i} = x_{\omega_i} + x_{s_i}$, where $x_{\overline{Q}_i}$ is the x-coordinate of the center of the square Q_i. The function Rec on the other hand implies $x_{v_i} = x_{\omega'_i} + x_{s_i}$. Subtracting the two equations we get:

$$|x_{\overline{Q}_i} - x_{v_i}| = |x_{\omega_i} - x_{\omega'_i}| \leq t < \frac{ka}{2} \Rightarrow x_{\overline{Q}_i} - \frac{ka}{2} < x_{v_i} < x_{\overline{Q}_i} + \frac{ka}{2}$$

and hence $x_{v_i} \in (x_{\overline{Q}_i} - \frac{ka}{2}, x_{\overline{Q}_i} + \frac{ka}{2})$. Similarly, for the y-coordinates we get $y_{v_i} \in (y_{\overline{Q}_i} - \frac{ka}{2}, y_{\overline{Q}_i} + \frac{ka}{2})$. Then, the point v_i constructed in the Rec function lies in the correct square and so the correct center $\overline{Q}_i$ can be identified. In other words, setting $x_{z_i} = x_{\overline{Q}_i} - x_{v_i}$ and $y_{z_i} = y_{\overline{Q}_i} - y_{v_i}$ implies that $z_i = \omega_i$.

Now assuming that $\mathsf{dist}(\boldsymbol{\omega}, \boldsymbol{\omega}') > t$ for at least one $i \in \{1, \dots, n\}$, by Definition 4 we have $|x_{\omega_i} - x_{\omega'_i}| > t$ or $|y_{\omega_i} - y_{\omega'_i}| > t$ (or both). Without loss of generality, we assume that $|x_{\omega_i} - x_{\omega'_i}| > t$. Using the equations $x_{\overline{Q}_i} = x_{\omega_i} + x_{s_i}$ and $x_{v_i} = x_{\omega'_i} + x_{s_i}$, we get that one of the following holds:

$$x_{\omega_i} - x_{\omega'_i} > t \Rightarrow x_{\omega'_i} < x_{\omega_i} - t \Rightarrow x_{v_i} < x_{\overline{Q}_i} - t$$
$$x_{\omega_i} - x_{\omega'_i} < -t \Rightarrow x_{\omega'_i} > x_{\omega_i} + t \Rightarrow x_{v_i} > x_{\overline{Q}_i} + t$$

Then $x_{v_i} \notin [x_{\overline{Q}_i} - t, x_{\overline{Q}_i} + t]$, and the reconstruction fails. Consequently, we obtain $x_{z_i} = x_{\overline{Q}_i} + x_{s_i} \neq x_{\omega_i}$, which means that the correct ω_i and hence $\boldsymbol{\omega}$ cannot be derived. □

Theorem 2. We work on the setting described in Section 3.2, assuming a grid G_a consisting of $n = N \times N$ squares and that an adversary possesses a sketch vector $\boldsymbol{s} = (s_1, \dots, s_n)$, where $s_i = (x_{s_i}, y_{s_i}) \in \mathbb{Z}^2$. Recall that each s_i in the sketch vector is the movement of a point $\omega_i = (x_{\omega_i}, y_{\omega_i}) \in \mathbb{Z}^2$ in a robust template $\boldsymbol{\omega} = (\omega_1, \dots, \omega_n)$ towards the closest center $\overline{Q}_i = (x_{\overline{Q}_i}, y_{\overline{Q}_i})$ of the square Q_i in the grid. In order for an adversary to obtain the secret point ω_i, it is sufficient to identify the square Q_i in which ω_i belongs to and compute $x_{\omega_i} = x_{\overline{Q}_i} - x_{s_i}$, $y_{\omega_i} = y_{\overline{Q}_i} - y_{s_i}$. Assuming that each ω_i is uniformly distributed in G_a, the best strategy for the adversary is to guess the square in which ω_i lies.

Let $X, Y \sim \mathrm{U}[-\frac{kaN}{2}, \frac{kaN}{2}]$ be discrete uniform random variables describing the coordinates of a point ω_i. The probability of choosing the correct point ω_i at random is given by the joint probability $\Pr(X = x_{\omega_i}, Y = y_{\omega_i}) = \frac{1}{(kaN)^2}$. This suggests that the joint min-entropy of X, Y is $H_\infty(X, Y) = 2\log_2(kaN)$. Since there are n points in $\boldsymbol{\omega}$, we conclude that the min-entropy of the robust template $\boldsymbol{\omega}$ is $m = 2n\log_2(kaN)$.

For simplicity we consider the two joint events $A_i = (X = x_{\omega_i}, Y = y_{\omega_i})$ and $B_i = (S = x_{s_i}, T = y_{s_i})$. Because the adversary already knows the sketch vector $\boldsymbol{s}$, we need to calculate to probability of guessing the point $\omega_i = (x_{\omega_i}, y_{\omega_i})$, given the corresponding sketch $s_i = (x_{s_i}, y_{s_i})$. This is the conditional probability:

$$\mathrm{P} = \Pr(A_i|B_i) = \Pr(B_i|A_i)\Pr(A_i) / \sum_{j=1}^{n} \left[\Pr(B_i|A_j)\Pr(A_j)\right]$$

where

$$\Pr(B_i|A_i) = \begin{cases} 1, & \text{if } \begin{cases} |x_{\omega_i} - x_{\overline{Q}_i}| < \frac{ka}{2} \text{ and} \\ |y_{\omega_i} - y_{\overline{Q}_i}| < \frac{ka}{2} \end{cases} \\ \frac{1}{2}, & \text{if } \begin{cases} |x_{\omega_i} - x_{\overline{Q}_i}| = \frac{ka}{2} \text{ or} \\ |y_{\omega_i} - y_{\overline{Q}_i}| = \frac{ka}{2} \end{cases} \end{cases}$$

is the probability of guessing the movement s_i of a point ω_i, when ω_i is given. Therefore, we distinguish two cases when computing the probability P, based on whether the given point ω_i lies in a square or on its border. In particular, we can verify that:

1. If $|x_{\omega_i} - x_{\overline{Q}_i}| < \frac{ka}{2}$ and $|y_{\omega_i} - y_{\overline{Q}_i}| < \frac{ka}{2}$, then:
$$\mathrm{P} = \frac{1 \times \frac{1}{(kaN)^2}}{\frac{1}{(kaN)^2} \sum_{j=1}^{n} \Pr(B_i|A_j)} = \frac{1}{n} = \frac{1}{N^2}$$
2. If $|x_{\omega_i} - x_{\overline{Q}_i}| = \frac{ka}{2}$ or $|y_{\omega_i} - y_{\overline{Q}_i}| = \frac{ka}{2}$, then:
$$\mathrm{P} = \frac{\frac{1}{2} \times \frac{1}{(kaN)^2}}{\frac{1}{(kaN)^2} \sum_{j=1}^{n} \Pr(B_i|A_j)} = \frac{1}{n} = \frac{1}{N^2}$$

Thus, in both cases, we get that $\Pr(A_i|B_i) = 1/N^2$. Then we can compute the average min-entropy of X, Y, given S, T as:

$$\tilde{H}_\infty(X, Y|S, T) = -\log_2 \left(\mathbb{E}_{(x_{s_i}, y_{s_i}) \leftarrow (S,T)} \left[\max_{x_{\omega_i}, y_{\omega_i}} \{\Pr(A_i|B_i)\} \right] \right)$$
$$= -\log_2 \frac{1}{N^2} = 2\log_2 N.$$

Because the vector $\boldsymbol{\omega}$ has n elements, the average min-entropy of $\boldsymbol{\omega}$ given the sketch vector $\boldsymbol{s}$ is $\tilde{m} = 2n\log_2 N$. In addition, the entropy loss is calculated via $m - \tilde{m} = 2n\log_2(kaN) - 2n\log_2 N = 2n\log_2(ka)$ and the required storage for the sketch vector $\boldsymbol{s}$ is $2n(\lfloor \log_2 ka \rfloor + 1)$. □

References

Adler, A., Youmaran, R., Loyka, S., 2009. Towards a measure of biometric feature information. Pattern Anal. Appl. 12 (3), 261–270. http://dx.doi.org/10.1007/S10044-008-0120-3.

Al-Assam, H., Abboud, A.J., Sellahewa, H., Jassim, S., 2012. Exploiting relative entropy and quality analysis in cumulative partial biometric fusion. Trans. Data Hiding Multim. Secur. 8, 1–18. http://dx.doi.org/10.1007/978-3-642-31971-6_1.

Arenas, M.P., Bingöl, M.A., Demirci, H., Fotiadis, G., Lenzini, G., 2022a. A secure authentication protocol for cholesteric spherical reflectors using homomorphic encryption. In: Batina, L., Daemen, J. (Eds.), Progress in Cryptology - AFRICACRYPT 2022: 13th International Conference on Cryptology in Africa, AFRICACRYPT 2022, Fes, Morocco, July 18-20, 2022, Proceedings. In: Lecture Notes in Computer Science, Springer Nature Switzerland, pp. 425–447. http://dx.doi.org/10.1007/978-3-031-17433-9_18.

Arenas, M., Demirci, H., Lenzini, G., 2021. Cholesteric spherical reflectors as physical unclonable identifiers in anti-counterfeiting. In: The 16th Int. Conf. on Availability, Reliability and Security. ACM, Vienna, pp. 1–11. http://dx.doi.org/10.1145/3465481.3465766, URL https://dl.acm.org/doi/10.1145/3465481.3465766.

Arenas, M., Demirci, H., Lenzini, G., 2022b. An analysis of cholesteric spherical reflector identifiers for object authenticity verification. Mach. Learn. Knowl. Extract. 4 (1), 222–239. http://dx.doi.org/10.3390/make4010010, URL https://www.mdpi.com/2504-4990/4/1/10.

Barker, E., Roginsky, A., 2019. Transitioning the Use of Cryptographic Algorithms and Key Lengths. NIST Special Publication 800-131A Revision 2 Tech. Rep. March, National Institute of Standards and Technology, Gaithersburg, MD, p. 33. http://dx.doi.org/10.6028/NIST.SP.800-131Ar2, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf.

Blanchet, B., 2001. An efficient cryptographic protocol verifier based on prolog rules. In: Proc. 14th IEEE Computer Security Foundations Workshop, vol. 1, IEEE, Citeseer, Nova Scotia, pp. 82–96.

Boyen, X., Dodis, Y., Katz, J., Ostrovsky, R., Smith, A., 2005. Secure remote authentication using biometric data. In: Cramer, R. (Ed.), In: Advances in Cryptology – EUROCRYPT 2005, vol. 3494, Springer Berlin Heidelberg, pp. 147–163. http://dx.doi.org/10.1007/11426639_9, URL http://link.springer.com/10.1007/11426639_9, Series Title, Lecture Notes in Computer Science.

Canetti, R., Fuller, B., Paneth, O., Reyzin, L., Smith, A., 2021. Reusable fuzzy extractors for low-entropy distributions. J. Cryptology 34 (1), 2.

Deledalle, C.-A., Denis, L., Tupin, F., 2012. How to compare noisy patches? Patch similarity beyond Gaussian noise. Int. J. Comput. Vis. 99 (1), 86–102. http://dx.doi.org/10.1007/s11263-012-0519-6, https://hal-imt.archives-ouvertes.fr/hal-00672357 http://link.springer.com/10.1007/s11263-012-0519-6.

Delvaux, J., Gu, D., Verbauwhede, I., Hiller, M., Yu, M.D.M., 2016. Efficient fuzzy extraction of PUF-induced secrets: Theory and applications. In: LNCS, vol. 9813 LNCS, pp. 412–431. http://dx.doi.org/10.1007/978-3-662-53140-2_20.

Dodis, Y., Kanukurthi, B., Katz, J., Reyzin, L., Smith, A., 2012. Robust fuzzy extractors and authenticated key agreement from close secrets. IEEE Trans. Inform. Theory 58 (9), 6207–6222. http://dx.doi.org/10.1109/TIT.2012.2200290.

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

Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.D., 2008. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM J. Comput. 38 (1), 97–139. http://dx.doi.org/10.1137/060651380.

Dodis, Y., Reyzin, L., Smith, A., 2004. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In: Int. Conf. on the Theory and Applications of Cryptographic Techniques. Springer, Switzerland, pp. 523–540.

Dolev, D., Yao, A., 1983. On the security of public key protocols. IEEE Trans. Inform. Theory 29 (2), 198–208.

Gedraite, E.S., Hadad, M., 2011. Investigation on the effect of a Gaussian blur in image filtering and segmentation. In: Proc. ELMAR-2011. IEEE, Switzerland, pp. 393–396.

Geng, Y., Noh, J., Drevensek-Olenik, I., Rupp, R., Lenzini, G., Lagerwall, J.P., 2016. High-fidelity spherical cholesteric liquid crystal Bragg reflectors generating unclonable patterns for secure authentication. Sci. Rep. 6, 1–9. http://dx.doi.org/10.1038/srep26840.

Jiang, Q., Zhang, N., Ni, J., Ma, J., Ma, X., Choo, K.-K.R., 2020. Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles. IEEE Trans. Veh. Technol. 69 (9), 9390–9401.

Kang, H., Hori, Y., Katashita, T., Hagiwara, M., Iwamura, K., 2014. Cryptographie key generation from PUF data using efficient fuzzy extractors. In: Proc. of 16th Int. Conf. on Advanced Communication Technology. ICACT, IEEE, Pyeongchang, Korea (South), pp. 23–26. http://dx.doi.org/10.1109/ICACT.2014.6778915.

Lenzini, G., Ouchani, S., Roenne, P., Ryan, P.Y., Geng, Y., Lagerwall, J., Noh, J.H., 2017. Security in the shell: An optical physical unclonable function made of shells of cholesteric liquid crystals. In: 2017 IEEE Workshop on Information Forensics and Security. WIFS 2017, vol. 2018-Janua, pp. 1–6. http://dx.doi.org/10.1109/WIFS.2017.8267644.

Li, N., Guo, F., Mu, Y., Susilo, W., Nepal, S., 2017. Fuzzy extractors for biometric identification. In: 2017 IEEE 37th Int. Conf. on Distributed Computing Systems. ICDCS, IEEE, United States, pp. 667–677.

Li, Q., Sutcu, Y., Memon, N., 2006. Secure sketch for biometric templates. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 4284 LNCS, pp. 99–113. http://dx.doi.org/10.1007/11935230_7.

Liu, X., Wang, Y., Li, X., Yi, Z., Deng, R., Liang, L., Xie, X., Loong, D.T., Song, S., Fan, D., All, A.H., Zhang, H., Huang, L., Liu, X., 2017. Binary tempor al upconversion codes of Mn^{2+}-activated nanoparticles for multilevel anti-counterfeiting. Nature Commun. 8 (1), 1–7. http://dx.doi.org/10.1038/s41467-017-00916-7.

Mesaritakis, C., Akriotou, M., Kapsalis, A., Grivas, E., Chaintoutis, C., Nikas, T., Syvridis, D., 2018. Physical unclonable function based on a multi-mode optical waveguide OPEN. Sci. Rep. 8 (1), 1–12. http://dx.doi.org/10.1038/s41598-018-28008-6.

Nandakumar, K., Jain, A.K., 2015a. Biometric template protection: Bridging the performance gap between theory and practice. IEEE Signal Process. Mag. 32 (5), 88–100.

Nandakumar, K., Jain, A.K., 2015b. Biometric template protection: Bridging the performance gap between theory and practice. IEEE Signal Process. Mag. 32 (5), 88–100. http://dx.doi.org/10.1109/MSP.2015.2427849.

Nisan, N., Zuckerman, D., 1996. Randomness is linear in space. J. Comput. System Sci. 52 (1), 43–52. http://dx.doi.org/10.1006/jcss.1996.0004.

Peng, W.H., Lee, M.Y., Li, T.H., Huang, C.H., Lin, P.C., 2016. Performance comparison of image keypoint detection, description, and matching methods. In: Proc. of IEEE 5th Global Conference on Consumer Electronics. GCCE 2016, IEEE, Kyoto, Japan, pp. 4–5. http://dx.doi.org/10.1109/GCCE.2016.7800416.

Schwartz, M., Geng, Y., Agha, H., Kizhakidathazhath, R., Liu, D., Lenzini, G., Lagerwall, J.P.F., 2021. Linking physical objects to their digital twins via fiducial markers designed for invisibility to humans. Multifunct. Mater. http://dx.doi.org/10.1088/2399-7532/ac0060, https://creativecommons.org/licences/by/3.0, https://iopscience.iop.org/article/10.1088/2399-7532/ac0060.

Shariati, S., Standaert, F.X., Jacques, L., Macq, B., 2012. Analysis and experimental evaluation of image-based PUFs. J. Cryptogr. Eng. 2 (3), 189–206. http://dx.doi.org/10.1007/s13389-012-0041-3.

Takahashi, K., Murakami, T., 2013. A measure of information gained through biometric systems. Image Vis. Comput. 32, http://dx.doi.org/10.1109/ICPR.2010.296.

Tuyls, P., Akkermans, A.H., Kevenaar, T.A., Schrijen, G.J., Bazen, A.M., Veldhuis, R.N., 2005. Practical biometric authentication with template protection. LNCS 3546, 436–446. http://dx.doi.org/10.1007/11527923_45.

Wang, Q., Wang, D., Cheng, C., He, D., 2021. Quantum2FA: Efficient quantum-resistant two-factor authentication scheme for mobile devices. IEEE Trans. Dependable Secure Comput. 20 (1), 193–208.

Wen, Y., Liu, S., 2018. Robustly reusable fuzzy extractor from standard assumptions. In: LNCS, vol. 11274 LNCS, Springer International Publishing, Brisbane, pp. 459–489. http://dx.doi.org/10.1007/978-3-030-03332-3_17.

Wen, Y., Liu, S., Gu, D., 2019. Generic constructions of robustly reusable fuzzy extractor. In: LNSC, vol. 11443 LNCS, Springer International Publishing, Beijing, pp. 349–378. http://dx.doi.org/10.1007/978-3-030-17259-6_12.

Youmaran, R., Adler, A., 2012. Measuring biometric sample quality in terms of biometric feature information in Iris images. J. Electr. Comput. Eng. 2012, 282589:1–282589:9. http://dx.doi.org/10.1155/2012/282589.

Zhu, H., Xiao, M., Sherman, D., Li, M., 2023. SoundLock: A novel user authentication scheme for VR devices using auditory-pupillary response. In: NDSS Symposium.

Dr. Mónica P. Arenas received her Ph.D. degree in Materials Science from the Federal University of Rio de Janeiro (Brazil). In 2020, she joined the Interdisciplinary Research Group in Sociotechnical Cybersecurity (IRiSC), as a postdoctoral researcher. Her main research interests include applied cryptography, biometric-like systems, data analytics and data privacy.

Dr. Georgios Fotiadis received his Ph.D. in Cryptography from the Department of Information and Communication Systems Engineering of the University of the Aeagean (Greece) in 2017. In 2019 he joined the Applied Security and Information Assurance (APSIA) group at SnT, University of Luxembourg, as a Research Associate. His current research interests are on the construction of secure cryptographic primitives, classical and post-quantum, focusing especially on elliptic curve-based primitives.

Prof.Dr. Gabriele Lenzini, Associate professor at the University of Luxembourg and Chief Scientist II at the University's Center for Reliability, Security and Trust (SnT), is head of the Interdisciplinary Research Group in Sociotechnical Security (IRiSC). Lenzini has more than 15 years of experience in the design and analysis of secure and private systems, a topic addressed using toolkits of different research methods, including formal and experimental analysis. In some of his most recent research, Lenzini focused on the problem of anti-counterfeiting, object authentication and the design of security protocols for that purpose.

Mohammadamin Rakeei received his M.Sc. degree in Secure Communications and Cryptography from Shahid Beheshti University (Iran), in 2020. His research interests include the design and analysis of security protocols, formal methods, provable security and network security. Since 2021, he is working as a doctoral researcher in the Interdisciplinary Research Group in Socio-technical Security, IRiSC, at Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg.