© Springer Berlin / Heidelberg, Lecture Notes in Computer Science (2026)

Fragmentation for Iris Template Protection

S. De Capitani di Vimercati[1], R. Donida Labati[1], S. Foresti[1], S. Jajodia[2], V. Piuri[1], P. Samarati[1], F. Scotti[1], and M. Vulpio[1]

[1] Università degli Studi di Milano, 20133 Milano, Italy
firstname.lastname@unimi.it
[2] George Mason University, Fairfax, VA 22030-4444, USA
jajodia@gmu.edu

Abstract. Existing solutions for protecting biometric templates are typically based on encryption or feature transformations, which may increase system's complexity or degrade its accuracy. In this paper, we put forward the idea of departing from encryption and protecting biometrics through fragmentation. By partitioning templates into unlinkable fragments, fragmentation provides protection since the limited information contained in each fragment is not sufficient for correct identification and hence cannot be exploited to impersonate legitimate subjects. In storage, fragments are mapped to fragtemplates on which the matching process is then performed. In each fragtemplate, genuine fragments are confused with synthetic ones and are indistinguishable from them. We design a fragmentation approach for iris templates and experimentally evaluate its accuracy and protection guarantees. Our experimental results confirm that fragmentation, when properly designed, can effectively provide protection while preserving identification accuracy.

Keywords: Biometric template protection · fragmentation

1 Introduction

Biometric-as-a-Service (BaaS) solutions have seen widespread adoption in recent years, and are increasingly used in high-assurance and large-scale identity scenarios as well as in everyday applications. Biometric information is inherently sensitive since identification and authentication are based on physical characteristics of each individual, which are typically neither revocable nor replaceable. In fact, although cancelable biometrics [21] solutions exist, they are not suitable for most real-world practical applications, due to their typical reduction in accuracy. The critical nature of biometric data is also reflected in regulatory frameworks (e.g., EU GDPR and EU AI Act [12, 13]), which classify such data as critical and impose strict requirements on their storage and use. Classical approaches for biometric data protection mainly rely on cryptographic mechanisms, which typically entail computational and/or communication overhead, or feature transformations, which may affect recognition accuracy [1].

In this paper, we put forward and explore the use of fragmentation for protecting biometric templates for their storage while departing from encryption.

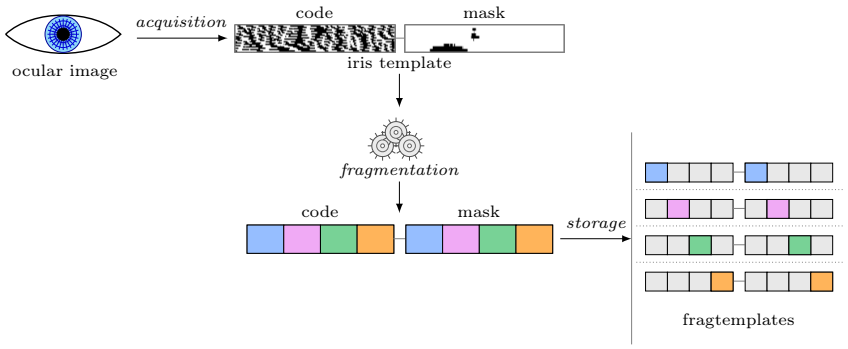


Fig. 1: Our approach

The basic idea is to partition each template in different fragments that are not linkable and, containing only a limited portion of the template, do not compromise its protection. Our work takes inspiration from fragmentation-based approaches that have been successfully investigated for data outsourcing [2]. To the best of our knowledge its application to biometric templates while departing from encryption has never been investigated. Such context brings new challenges with respect to classical relational data management scenarios. For biometric data, or more precisely biometric templates, what is to be maintained protected is the template itself. Such protection must guarantee that, even if fragments are improperly leaked or accessed, they will not expose the template and attempts of using such fragments for masquerading as the original identity will be ineffective. At the same time, our goal is to perform the biometric recognition process directly on fragments, in contrast to retrieving and recombining the fragments and reconstructing the original template for matching (which would clearly imply computational overhead). Therefore, there are two contrasting requirements to balance: on one side the *confidentiality* of the original templates, and on the other side the *accuracy* of the biometric matching process when executed on fragments.

Our approach. For concreteness, in this paper, we consider iris templates, that is, the representation of the biometry obtained from ocular images (Section 2). The reason for this is that iris template matching, being based on normalized Hamming distances, can leverage the semi-additive nature of such measure conveniently enabling to operate the matching on fragments. We operate an aligned fragmentation on the two components (code and mask) of iris templates, ensuring that each fragment code has a corresponding fragment mask, resulting in a template fragment.

The basic protection of fragmentation is the fact that each fragment provides only limited information not sufficient for exploiting the fragment for identification. In addition, our approach further protects fragments by hiding them in fragtemplates (Section 3). A fragtemplate contains one genuine fragment while all the remaining fragments in it are synthetic. Fragtemplates have the same

structure and size as the genuine iris templates, and the genuine fragment within them remains indistinguishable from the others, and not recognizable as such by parties observing stored fragments. Fragtemplates corresponding to a genuine template are all associated with a different identifier and stored at different machines (i.e., external servers). Figure 1 illustrates the basic idea of our approach (colored fragments are genuine, gray fragments are synthetic).

To perform a matching of a query template against datasets of fragtemplates, the query template is fragmented: query fragments are sent to the storing servers for matching, and the results against the different fragtemplates are combined (Section 4). As common in similar scenarios (e.g., [6]), we assume that parties storing fragtemplates and performing the matching on fragtemplates are trusted to correctly store templates and execute the matching process requested to them, and that communication to/from them is protected by classical approaches. The effectiveness of the proposed approach is experimentally evaluated with respect to both identification accuracy and template protection (Section 5).

2 Basic concepts

An iris template is obtained from an ocular image which, through filtering and elaboration (enrollment) [9], is codified as a pair $\langle C,M\rangle$ of binary matrices, where C is the *iris Code* and M its *segmentation Mask*. Iris code C represents the iris texture [28], with rows corresponding to radial samples (concentric iris bands) and columns corresponding to angular samples of the eye. Segmentation mask M (mask, for short) indicates which bits of C are reliable and should be considered, and which should be excluded (e.g., bits corresponding to noisy or occluded regions) when matching the iris template with other iris templates. Basically, only entries $C[i,j]$ for which $M[i,j]$ has value 1 are considered.

The match between two iris templates, typically between a new acquisition $\langle \mathtt{C},\mathtt{M}\rangle$ (*query template*) and a stored iris template $\langle C,M\rangle$ obtained during enrollment, is based on the *normalized Hamming Distance* (HD) between the two. Formally, $\mathrm{HD} = \frac{\|(\mathtt{C}\oplus C)\odot(\mathtt{M}\odot M)\|_1}{\|\mathtt{M}\odot M\|_1}$, where $\oplus$ is the element-wise exclusive OR (XOR) operator, $\odot$ is the element-wise (Hadamard) product, and $\|\cdot\|_1$ is the ℓ_1 norm and counts the number of non-zero elements. In other words, HD measures the ratio of bits that are reliable in both templates (i.e., both masks have value 1) and for which their iris codes differ. The smaller the HD, the higher the similarity between the templates.[3] Template matching can be used in a 1:1 (*authentication*) or a 1:n (*identification*) scenario. The first matches a query template against a specific reference template, returning success if the distance is below a given threshold. The second matches a query template against a set

[3] Due to head pose variations and eye rotation during acquisition, the alignment between two iris templates is generally unknown. Such rotational misalignments are modeled by circular shifts of the iris code and mask along the angular dimension, corresponding to horizontal shifts of the matrix columns [9]. HD must therefore be computed for all considered shifts of the templates being matched, considering then the minimum distance over such shifts.

of reference templates, each associated with its identifier, returning the identifier of the reference template with the smallest distance, that is, the most similar.

For concreteness, but without loss of generality, in this paper we consider the problem of 1:n matching, as the ability to support it subsumes the ability to support the 1:1 scenario.

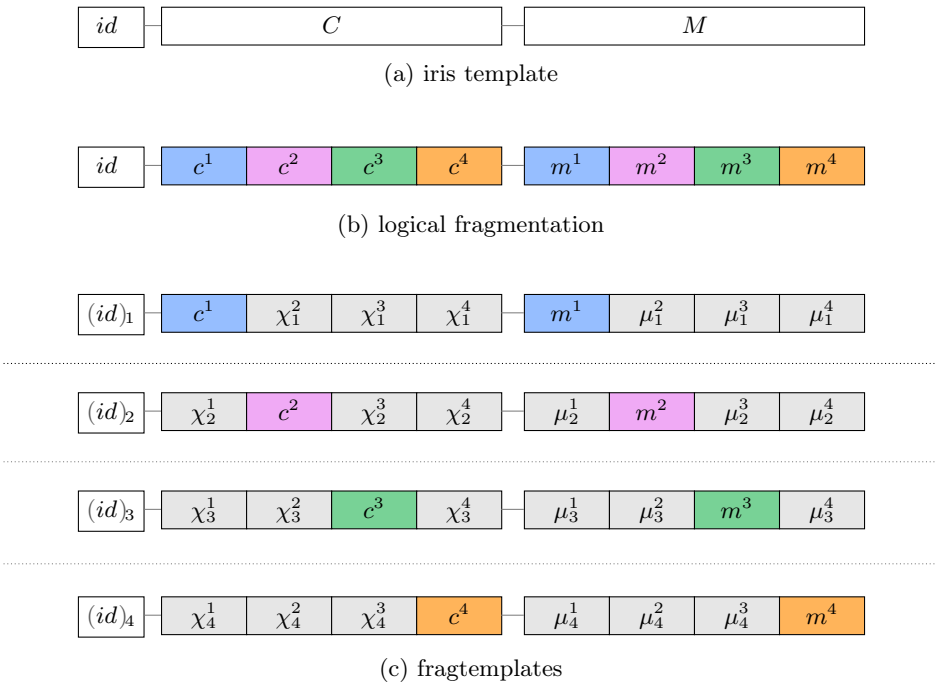
3 Iris template fragmentation

In this section, we first introduce the fragmentation process (Section 3.1), and then the construction of fragtemplates (Section 3.2).

3.1 Fragments definition

Fragmenting an iris template means partitioning the matrices of its iris code and mask, separating their cells into different sets that are disjoint and together fully cover the original matrices. In principle, any partitioning of the matrices would do and a different partitioning could be applied to the iris code and the mask of a template. However, a random partitioning, while providing support for template confidentiality, would suffer in terms of functionality. Not only would it imply an overhead for maintaining fragmentation maps at the client, but it would produce fragments that are not able to easily support matching. To support matching at the level of fragment, fragments should cover contiguous areas of the iris code (and of the corresponding mask), that is, there must not be missing bits within the matrix area covered by a fragment. This requirement arises from the fact that matching relies on comparing corresponding positions in the iris code and mask. Formally, $\forall i, j, l, m$, if $C[i, j]$ and $C[l, m]$, with $i \leq l, j \leq m$ belong to the same fragment then, $\forall x, y : i \leq x \leq l, j \leq y \leq m$, $C[x, y]$ belongs to that fragment. The same applies to M. Trivially, this can be obtained with a partitioning that operates by splitting the matrix either *vertically*, that is, by columns (i.e., angular fragments of the iris image), or *horizontally*, that is, by rows (i.e., radial fragments of the iris image). In both cases, fragments are disjoint and together cover the entire template. Also, to execute matching of fragments, the fragmentation of an iris template must be performed consistently on both C and M, meaning that the same rows or columns must be grouped together in both matrices.

To operate agnostically with respect to horizontal or vertical fragmentation, we model the binary matrices C and M as two binary vectors obtained by reading the matrices either by rows or by columns, depending on the fragmentation strategy. In fact, the horizontal (vertical, resp.) fragmentation of a $n_{row} \times n_{col}$ binary matrix corresponds to a partitioning of the $n_{row} \cdot n_{col}$ binary vector obtained by reading the matrix by rows (columns, resp.), considering as split points only multiples of n_{row} (n_{col}, resp.). Splitting the vector at such positions ensures that contiguous rows (columns, resp.) are kept together in the same fragment. In the following, we use C and M to denote either the matrices or the corresponding vectors. Formally, the fragmentation of an iris template is defined as follows.



(a) iris template

(b) logical fragmentation

(c) fragtemplates

Fig. 2: An example of iris template (a), its logical fragmentation in four fragments of the same size (b), and the corresponding fragtemplates (c)

Definition 1 (Logical fragmentation of an iris template). *Let $\langle C, M\rangle$ be an iris template, with C and M the binary vectors representing the iris code and the mask, respectively. A* logical fragmentation *of the iris template in f fragments is $\langle c^1, \ldots, c^f, m^1, \ldots, m^f\rangle$, where $\langle c^1, \ldots, c^f\rangle$ and $\langle m^1, \ldots, m^f\rangle$ are partitions of C and M, respectively, such that:*

1. $C = \|_{i=1}^{f} c^i$, $M = \|_{i=1}^{f} m^i$, *with $\|$ the concatenation operator (partitioning);*
2. $|c^i| = |m^i|$, $i = 1, \ldots, f$ *(consistent iris code/mask fragmentation);*
3. $|c^i| = |c^j|$, $i, j = 1, \ldots, f$ *(uniform size).*

Condition 1 ensures that fragments are disjoint and that, all together, cover the entire iris template. Combined with Condition 2, which requires each fragment of the iris code and of the mask to have the same size, these conditions guarantee consistency in the partitioning of the iris code and its mask, so that their fragments can be aligned. (i.e., every iris code fragment has a corresponding mask fragment). Finally, Condition 3 ensures that all fragments have the same size. This condition is not necessary for protection or functionality, but it simplifies treatment and provides convenience in the outsourcing of the matching process. Figures 2(a)-(b) show a simple example of an iris template and its

logical fragmentation, considering $f = 4$ fragments. In the figure, we use background colors to distinguish among fragments, and the color-coding gives the correspondence between fragments in the iris code and in the mask (i.e., covering the same positions in the two vectors). The *id* is reported associated with the template, we will comment on this in the next section.

3.2 Fragtemplates

When applying fragmentation to a dataset of iris templates, each fragment is combined with fragments of synthetic iris templates to obtain a template, which we call *fragtemplate*, that has the same format and size as a genuine iris template. The logical fragmentation of an iris template is then mapped to a set of fragtemplates, one for each fragment in the logical fragmentation. Each fragtemplate is a partially synthetic iris template having the same size as a genuine iris template. More specifically, each i-th fragment in the logical fragmentation is mapped to a fragtemplate that, for both the iris code and the mask, has the genuine bits at the positions corresponding to c^i (m^i, resp.), while the remaining bits are synthetic. Synthetic (codes and masks) fragments are extracted from synthetic iris templates generated so as to mimic the statistical properties of genuine data [33], avoiding patterns that could reveal their synthetic nature. We maintain such indistinguishability in their application by ensuring that each synthetic fragment (extracted from synthetic template) is used only once. Also, in each fragtemplate, fragments derive from different synthetic templates. The genuine fragment therefore remains indistinguishable from the others, and not recognizable as such.

Each enrolled iris template must be associated with the identity *id* of the subject from which the template was acquired. In the following, with a slight abuse of terminology, we use the term *iris template* to denote either the pair $\langle C, M\rangle$ or such pair in association with the identifier, that is, $[id, \langle C, M\rangle]$ depending on the context. Mapping enrolled templates to fragtemplates requires maintaining the association of the fragtemplates with the identity, while ensuring that fragtemplates remain unlinkable. This implies that the same identifier cannot be used for them (identifiers associated with fragtemplates should all be different one from another). At the same time, it should be possible for the client to retrieve all identifiers to which a given subject *id* has been mapped. A convenient way to achieve these requirements is through the iterative application of keyed hashing (HMAC) [17]: iteratively applying a hash function h with a key *key* enables the derivation of a chain of identifiers. Starting from the original subject identifier *id*, the i-th identifier (to be associated with the i-th fragtemplate) is obtained by iteratively applying the keyed hash function i times. With a slight abuse of notation, we use h^i to denote the iterative application i times of the hash function. That is, $h^i(x,key) = h(h^{i-1}(x,key),key)$ and $h^1(x,key) = h(x,key)$. This approach enables the client to retrieve all identifiers while maintaining the unlinkability of the fragtemplates since, without knowledge of the *key*, it is not possible to reconstruct the chain of identifiers. We are now ready to formally define the concept of fragtemplate.

Definition 2 (Fragtemplate). *Let* $[id, \langle C, M\rangle]$ *be an enrolled iris template, with id the subject identifier,* C *and* M *the binary vectors of the iris code and mask, respectively, and* $\langle c^1, \ldots, c^f, m^1, \ldots, m^f\rangle$ *be its logical fragmentation. The* fragtemplate *corresponding to the* i*-th logical fragment* (c^i, m^i)*, with* $i = 1, \ldots, f$*, is a template* $[\overline{id}_i, \langle \overline{C}_i, \overline{M}_i\rangle]$ *such that:*

1. $\overline{id}_i = h^i(id,key)$*;*
2. $\overline{C}_i = \|_{j=1}^{f} \overline{c}_i^j$*, where* $\overline{c}_i^i = c^i$ *and* $\overline{c}_i^j$ *is synthetic for* $i \neq j$*;*
3. $\overline{M}_i = \|_{j=1}^{f} \overline{m}_i^j$*, where* $\overline{m}_i^i = m^i$ *and* $\overline{m}_i^j$ *is synthetic for* $i \neq j$*.*

Intuitively, with a logical fragmentation in f fragments, an enrolled template is mapped to f fragtemplates, one for each fragment in the logical fragmentation. The i-th fragtemplate, whose identifier is obtained by applying the keyed hash function i times on the original id, has the i-th fragment of the iris code (mask, respectively) corresponding to the genuine fragment while all the other fragments are synthetic. For simplicity but without loss of generality, we assume that the i-th fragment of the logical fragmentation is mapped to the i-th fragtemplate.

Fragtemplates can be used for identification since the client knows which fragment is genuine in each fragtemplate, and can reconstruct the chain of identifiers. On the contrary, without the key and the correspondence between logical fragments and fragtemplate, there is no easy way to distinguish genuine fragments or reconstruct the association among the fragtemplates of the same iris template. When applying fragmentation to a dataset $\mathcal{I}$ of iris templates, the same logical fragmentation (i.e., the same number of fragments and hence same partitioning of the vectors) must be applied to all the templates, so that their corresponding fragtemplates have all the same structure and size. In the following, we denote with $\mathcal{I}_i$ the dataset of the i-th fragtemplates of the iris templates in dataset $\mathcal{I}$. Figure 2(c) illustrates the four fragtemplates corresponding to the logical fragmentation in Figure 2(b). In the figure, we use Greek letters for synthetic fragment content. Note that while color-coding is provided in the figure to denote the genuine fragment, the fragment is unrecognizable as such and indistinguishable from the synthetic ones at the storage level.

4 Identification process

Our next problem is to translate template matching to work on fragtemplates. Given the semi-additive nature of the normalized Hamming distance, this operation is in principle relatively straightforward. In fact, the number of relevant bits and of bits XOR-ing to ones can be computed at the level of fragment and then combined to compute the distance at the template level. In this paper, we consider row-level fragmentation, which enables such computation while still accounting for rotational misalignments, since the required shifts of bits remain within each fragment.[4]

[4] Vertical fragmentation would permit a large number of fragments, but it requires a revision in the identification process for the support of rotational misalignments.

Main /* executed at client */
1: Let $\langle \mathtt{C}, \mathtt{M} \rangle$ be a query template
2: $\langle \mathtt{c}^1, \ldots, \mathtt{c}^f, \mathtt{m}^1, \ldots, \mathtt{m}^f \rangle := \mathtt{Fragment}(\langle \mathtt{C}, \mathtt{M} \rangle)$ /* fragment the query template */
3: **for** $j := 1$ **to** f **do** /* for each logical fragment */
4: $(R_j^1, \ldots, R_j^f) := \mathbf{Rank}(\mathtt{c}^j, \mathtt{m}^j, k)$ /* send j-th fragment to compute the rankings on $\mathcal{I}_j$ */
5: **for** $j := 1$ **to** f **do** /* for each rank computed on genuine fragments */
6: $R_j^j := h^{f-j}(R_j^j, key)$ /* transform subjects' identifiers to comparable values */
7: **return**($\mathtt{Rank\text{-}1}(\mathtt{Combine}(R_1^1, \ldots, R_f^f))$) /* combine rankings and return rank-1 */

Rank$(\mathtt{c}^j, \mathtt{m}^j, k)$ /* against dataset $\mathcal{I}_j$ */
1: Let $D(id, dist^1, \ldots, dist^f)$ be the distance table /* table storing computed HDs */
2: **for each** $[\overline{id}_x, \langle \overline{C}_x, \overline{M}_x \rangle] \in \mathcal{I}_j$ **do** /* for each fragtemplate in dataset $\mathcal{I}_j$ */
3: $d_x[id] := \overline{id}_x$ /* create a row in D for $\overline{id}_x$ */
4: **for** $i := 1$ **to** f **do** /* for each fragment in the x-th fragtemplate */
5: $d_x[dist^i] := \mathtt{HD}(\langle \mathtt{c}^j, \mathtt{m}^j \rangle, \langle \overline{c}_x^i, \overline{m}_x^i \rangle)$ /* compute HD between $\langle \mathtt{c}^j, \mathtt{m}^j \rangle$ and the i-th frag. */
6: **for** $i := 1$ **to** f **do** /* for each fragment */
7: $R^i := \mathtt{Top}\text{-}k(D[id], D[dist^i])$ /* compute the top-k subject identifiers with the smallest HD */
8: **return**$(R^1, \ldots, R^f)$ /* return the top-k rankings */

Fig. 3: Pseudocode of the identification process

Matching on fragments can be enforced by partitioning the query template in the same number of fragments as the enrolled templates, and by comparing each fragment of the query template with the corresponding genuine fragment of each enrolled template in the iris dataset. With reference to fragtemplates in which such fragments are hidden, this requires comparing the i-th fragment of the query template with the i-th fragment of the fragtemplates in dataset $\mathcal{I}_i$. Formally, each fragment $\langle \mathtt{c}^i, \mathtt{m}^i \rangle$, $i = 1, \ldots, f$, of the query template should be matched against each fragment $\langle \overline{c}_i^i, \overline{m}_i^i \rangle$ in the dataset $\mathcal{I}_i$ of the i-th fragtemplates. In fact, all the other fragments in the fragtemplates are synthetic and their purpose is only to provide protection for the genuine fragment in storage. However, performing matching only on the genuine fragment within a fragtemplate would disclose it as such (and disclose as synthetic those that are not queried). To avoid this, we perform matching on all fragments, and simply discard results returned for synthetic fragments. In other words, $\langle \mathtt{c}^i, \mathtt{m}^i \rangle$ is also compared against $\langle \overline{c}_i^j, \overline{m}_i^j \rangle$, with $j \neq i$, then ignoring the results of such matching.

Translating the matching to operate on fragtemplates, we also note two factors that can be leveraged to maintain processing at the client negligible. First, genuine fragments whose similarity with the query fragment is very low can be safely ignored. Second, operating on ranks of identifiers based on similarity proves sufficient for correct identification (Section 5). We leverage these observations by requesting the server storing dataset $\mathcal{I}_i$ to match query fragment $\langle \mathtt{c}^i, \mathtt{m}^i \rangle$, $i = 1, \ldots, f$, against its fragtemplates and to return the list of k identities (top-k) associated with the smallest Hamming distances, ranked in increasing order of HD. As a matter of fact, while not always the closest, the true result will

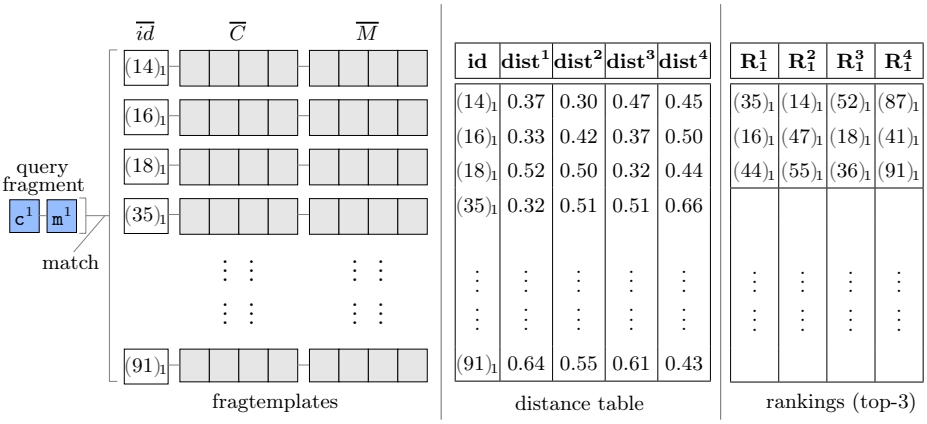


Fig. 4: An example of matching for query fragment $\langle \mathtt{c}^1, \mathtt{m}^1 \rangle$

typically fall in the top positions of the ranks locally computed at the fragment level. The result can then be obtained by combining ranks. In our experiments, majority voting (considering the identity appearing in the highest number of ranks), with Borda count, weighting position in the ranks in case of tie, proved to work well (see Section 5).

Before wrapping up the description of the matching through fragtemplates there is another note to make. Fragtemplates corresponding to the same subject are associated with different identifiers, since the identifier of each subject is represented in a different manner in each of its fragtemplates (see Definition 2). Also, since keyed hash functions are non-invertible, it is not possible to derive the identifier $\overline{id}_y$ of the y-th fragtemplate from the identifier $\overline{id}_x$ of the x-th fragtemplate when $y < x$, nor to recover the original identifier. It is, however, possible, knowing the key, to derive identifiers forward. To combine rankings, we then transform all rankings to operate on the domain of the identifiers of the f-th (last) fragment. More precisely, the identifiers of the x-th fragtemplates are transformed onto the domain of identifiers of the f-th fragtemplates by iteratively applying the keyed hash function $f - x$ times.

Figure 3 illustrates the pseudocode of the identification process on fragtemplates. It comprises a **main** (executed at the client) and a function **rank** (executed on each of the f fragtemplate datasets). Given a query template $\langle \mathtt{C}, \mathtt{M} \rangle$ the main process first logically partitions it into f fragments (line 2). Each of these fragments $\langle \mathtt{c}^j, \mathtt{m}^j \rangle$, $j = 1, \ldots, f$, is then sent to the servers (call to function **rank**) for matching against the dataset $\mathcal{I}_j$ corresponding to the j-th fragtemplates (i.e., the fragtemplates in which the j-th fragment is genuine). Such matching returns the rankings $R_j^1, \ldots, R_j^f$ of the top-k identities matching the input query frag-

$\mathbf{R_1^1}$	$\mathbf{R_1^2}$	$\mathbf{R_1^3}$	$\mathbf{R_1^4}$
$(35)_1$	$(14)_1$	$(52)_1$	$(87)_1$
$(16)_1$	$(47)_1$	$(18)_1$	$(41)_1$
$(44)_1$	$(55)_1$	$(36)_1$	$(91)_1$

$\mathbf{R_2^1}$	$\mathbf{R_2^2}$	$\mathbf{R_2^3}$	$\mathbf{R_2^4}$
$(12)_2$	$(39)_2$	$(32)_2$	$(33)_2$
$(23)_2$	$(35)_2$	$(28)_2$	$(34)_2$
$(27)_2$	$(26)_2$	$(72)_2$	$(09)_2$

$\mathbf{R_3^1}$	$\mathbf{R_3^2}$	$\mathbf{R_3^3}$	$\mathbf{R_3^4}$
$(25)_3$	$(82)_3$	$(94)_3$	$(75)_3$
$(13)_3$	$(10)_3$	$(35)_3$	$(61)_3$
$(80)_3$	$(61)_3$	$(83)_3$	$(85)_3$

$\mathbf{R_4^1}$	$\mathbf{R_4^2}$	$\mathbf{R_4^3}$	$\mathbf{R_4^4}$
$(12)_4$	$(51)_4$	$(18)_4$	$(35)_4$
$(91)_4$	$(33)_4$	$(43)_4$	$(70)_4$
$(11)_4$	$(60)_4$	$(51)_4$	$(55)_4$

Fig. 5: An example of rankings for the fragtemplates in Figure 2(c)

ment considering comparison with all fragments in each of the fragtemplates in the j-th dataset (lines 3-4). For each j-th fragment, only R_j^j (the rank on the genuine fragment) is considered, and identities in the different ranks are transformed to be on the domain of identities of the last dataset (lines 5-6). Finally, ranks are combined and the result (rank-1) returned (line 7).

Function **rank** operates independently on each fragtemplates dataset. Operating at dataset $\mathcal{I}_j$, it compares the input query fragment $\langle \mathtt{c}^j, \mathtt{m}^j \rangle$ with each fragment of the fragtemplates in $\mathcal{I}_j$, computing their normalized Hamming Distance. For convenience of the illustration, we assume distances to be represented as a table D with a column for the identifiers and f additional columns for the distances. For each fragtemplate in $\mathcal{I}_j$ (line 2), a tuple is added to D with the identifier of the fragtemplate and the distance between the query fragment and each of the fragments in it (lines 4-5). Finally, for each of the f columns in D, the identifiers corresponding to the smallest k values of such recorded distances are returned.

Figures 4 and 5 illustrate the matching process, continuing on the example with $f = 4$ fragments in Figure 2. For readability of the example, in the figures, the identities in fragtemplates are represented with the value of the original identifier in parenthesis and a subscript corresponding to the application of the keyed hash function. For instance, original identity 35, which would translate in the i-th fragment to $h^i(35,key)$, is represented as $(35)_i$.

Figure 4 shows the matching of the first query fragment with the dataset $\mathcal{I}_1$ of fragtemplates. The distance table in the middle has one tuple for each of such fragtemplates with the fragtemplate identifier in the first column and, in each column $dist^i$, the distance between query fragment $\langle \mathtt{c}^1, \mathtt{m}^1 \rangle$ and the i-th fragment in the fragtemplate. The table on the right reports, in each column R^i, the rank of the top-3 identities (i.e., $k = 3$) for which the i-th fragment has smallest distance from $\langle \mathtt{c}^1, \mathtt{m}^1 \rangle$.

Figure 5 shows the tables with top-3 ranks retrieved from the different servers. For each table, the ranks corresponding to the genuine fragment are color-coded (ranks in clear columns correspond to ranks computed on synthetic fragments and are to be ignored). Identity 35, appearing in all top-k ranks, and hence voted by the majority, is returned as result.

5 Experimental results

To assess the effectiveness of our approach, we implemented it and experimentally evaluated both identification accuracy and template protection.

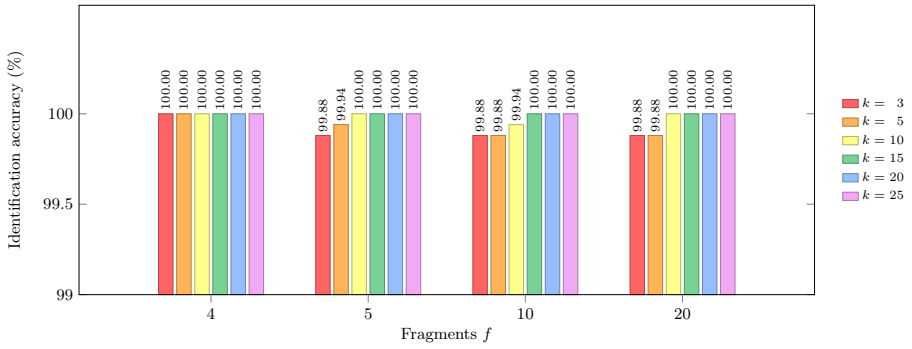


Fig. 6: Identification accuracy (%) varying the number f of fragments and the rank size k (baseline 100%)

5.1 Experimental libraries, settings, and datasets

We implemented our approach using publicly available software libraries. For feature extraction and matching, we used the University of Salzburg Iris Toolkit (USIT) [30]. Feature extraction has been performed using Logarithmic Gabor (LOG) filters [22]. We adopted the default USIT parameters, obtaining iris templates consisting of 20×512 binary matrices, where odd rows represent the quantized real part of the Log-Gabor filter responses and even rows represent the quantized imaginary part. The parameters approximating the inner and outer iris boundaries have been computed using the algorithm in [15]. For the experimental validation, we used the following datasets.

Iris dataset. We used a subset of the IIT Delhi Iris Database version 1.0 [18], which is a standard benchmark for iris recognition. It includes images acquired using commercial iris scanners under controlled conditions, with a resolution of 320×240 pixels. To guarantee high confidence in accuracy results, we maintained only the eyes with five different samples: we used one sample to populate the iris dataset, and four to simulate identification queries. Since our goal is to measure identification accuracy with fragmentation, we discarded those eyes for which the publicly available tools produced unreliable estimations of the circles approximating the inner and outer iris boundaries, hence comparing against a baseline of 100% accuracy when fragmentation is not used. Hence, our dataset included 415 eyes for which manually segmented masks are publicly available [14], corresponding to 2,075 samples.

Synthetic image dataset. We used the full dataset of 4,167 synthetic iris images of different individuals, without textured contact lenses proposed in [24]. These images were generated using a Generative Adversarial Network trained on images acquired with the same sensor and under acquisition conditions similar to the ones of our iris dataset. Since manually segmented masks are not publicly available for this dataset, we segmented the iris region using the publicly available software in [24].

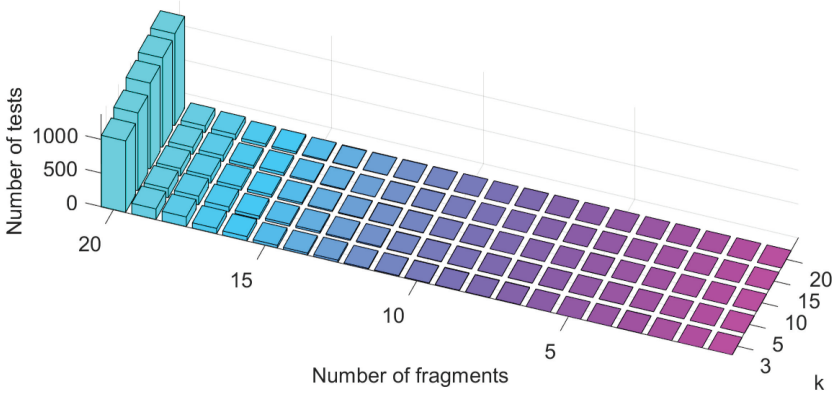


Fig. 7: Number of identification tests in which the correct result appears in the k rankings of $20, \ldots, 0$ fragments

5.2 Identification accuracy

We first focus on evaluating whether fragmentation affects identification accuracy, measured as the rank-1 identification accuracy [16], that is, the correctness of the identity returned in a 1:n matching. The results illustrated in the following have been obtained considering four identification attempts for each of the 415 subjects in the iris dataset, resulting in 1,660 closed-set identification tests.

Figure 6 illustrates the identification accuracy varying the number f of fragments in the set $\{4, 5, 10, 20\}$ (i.e., the divisors of the number of rows in an iris code discarding 2, which is considered too low), and the value k for top-k ranking in the set $\{3, 5, 10, 15, 20, 25\}$. Identification accuracy remains close to 100% in all the considered configurations. As visible from the figure, k has a limited impact on identification accuracy beyond small values: increasing k from 3 to 10 leads to minor improvements, while for $k \geq 15$ the accuracy saturates at 100% in all the tested configurations. When increasing f, which results in a finer partitioning of the iris templates with each fragment storing a smaller portion of the original template, accuracy is maintained at 100% increasing the value of k (3 is sufficient for $f = 4$, 10 is necessary for $f = 20$). The results then confirm that fragmentation has a negligible practical impact on identification accuracy, even for a large number of fragments.

The experiments also confirm that operating on ranks considering majority voting and using Borda count in case of ties, guarantees high identification accuracy. Figure 7 reports the number of identification tests in which the correct result appears in the top-k rankings for $f = 20$, varying k. The x-axis represents the number of fragments. As visible from the figure, in the majority of tests the result appeared in the top-k rankings of all the fragtemplates. Also, other identities in the top-k ranks had much lower occurrences, below 5 in almost all the cases (99.96%). Only in two cases, Borda was called to decide the tie.

5.3 Template protection

To evaluate the protection guarantees provided by fragmentation, we investigated whether a fragtemplate can be successfully used as a biometric credential in a traditional iris recognition system. The goal of the analysis was to determine a fragmentation level (f) such that the information contained in a fragtemplate is not sufficient for impersonation.

An iris recognition system grants access to a subject when the Hamming Distance (HD) between the query and an enrolled template is below a threshold and, if access is granted, identifies the subject as the one corresponding to the template with the smallest HD. The threshold is fixed by the system to properly separate the probability density functions of genuine and impostor matching scores in the dataset. The genuine matches are comparisons between samples belonging to the same subject, while the impostor matches are comparisons between samples belonging to different subjects. These scores typically form two separated bell-shaped distributions, whose relative position characterizes the discriminative capability of the system [8]. For our dataset, the threshold was set to 0.416, corresponding to the Equal Error Rate estimated for the reference identity verification system on the iris dataset used [9].

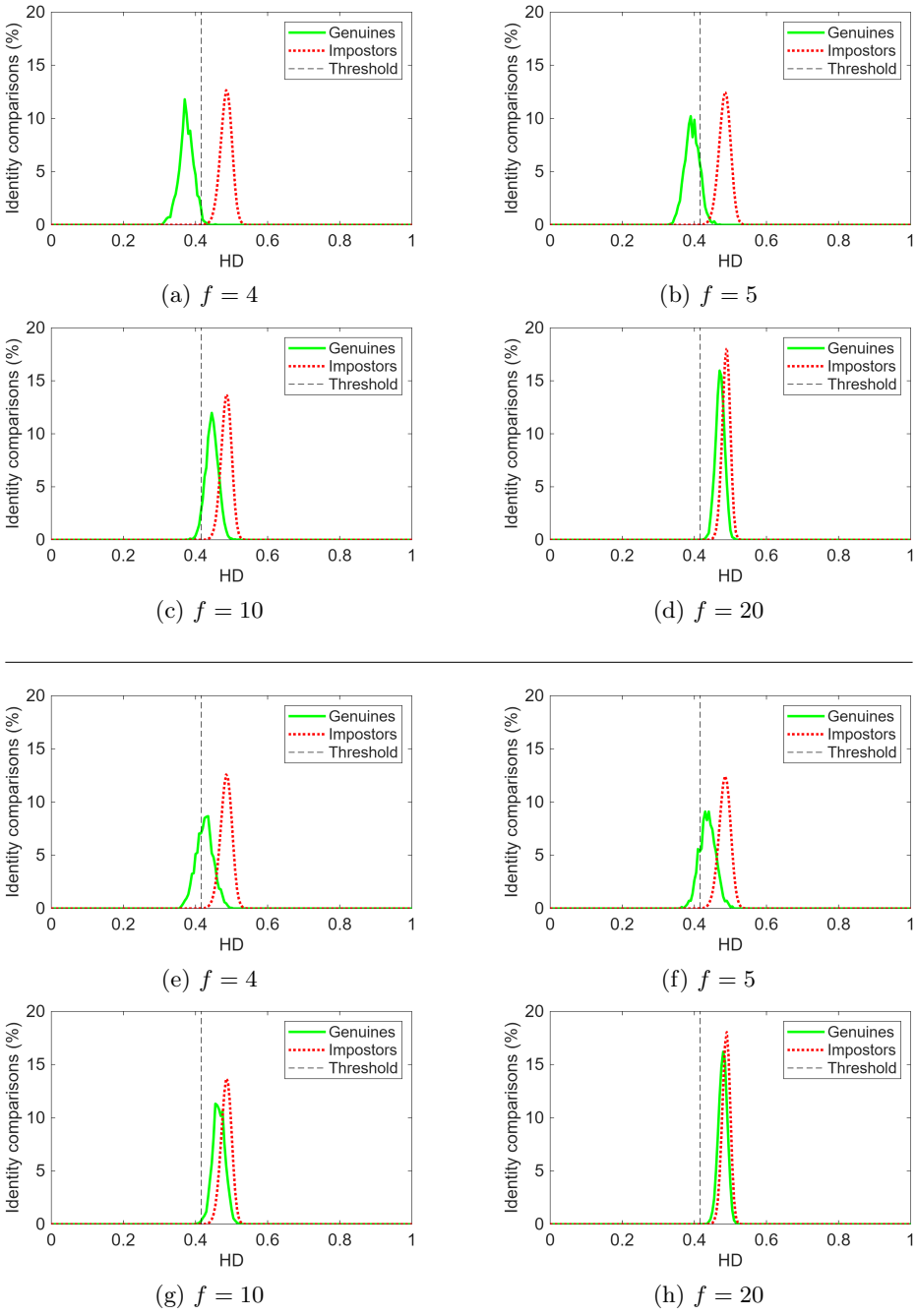
In our experiments, we assumed the worst possible scenario in which the external system includes exactly the same set of subjects as the original dataset that was fragmented. For the analysis, we considered two scenarios:

- *worst-case*: the enrolled templates in the external system are exactly the same as those that were fragmented (this is a theoretical scenario that is very unlikely to happen in practice);
- *realistic*: the enrolled templates in the external system, while from the same set of enrolled subjects have been obtained from a different acquisition.

In both scenarios, fragtemplates are used as biometric credentials to attempt access to the external iris recognition system. We then compared each fragtemplate in our datasets against each iris template in the external system for all configurations with f in the set $\{4, 5, 10, 20\}$, resulting, for each of them, in $172{,}225 \cdot f$ identity comparisons performed.

Figure 8 illustrates the probability density functions of the genuine match (continuous green line) and impostor match (dotted red line) score distributions varying the numbers f of fragments for both the worst-case (a-d) and the realistic scenario (e-h). A fragtemplate is successfully used as a biometric credential when it is accepted by the external iris recognition system. Given that a genuine match corresponds to the comparison of a fragtemplate with the whole genuine template of the same subject, the area of the green curve before the threshold corresponds to fragtemplates that have successfully passed the recognition. In other words, the larger the area of the green curve before the threshold, the greater the adversary's ability to use a fragtemplate for impersonation.

In the worst-case scenario, when $f = 4$, the area of the green curve is almost completely to the left of the threshold, signaling that fragtemplates are recognized as genuine by the external system and hence that fragmentation does



(a) $f = 4$ (b) $f = 5$

(c) $f = 10$ (d) $f = 20$

(e) $f = 4$ (f) $f = 5$

(g) $f = 10$ (h) $f = 20$

Fig. 8: Probability density functions of genuine and impostor scores for different values of f in the worst-case (a)-(d) and realistic scenarios (e)-(h)

not succeed in providing protection. This is clearly to be expected since $f = 4$ implies that each fragtemplate maintains one fourth of the original template (considerably lowering the overall HD in the match). As the number of fragments increases, the green curve moves to the right, implying an increase of protection. When $f = 20$, the green curve area lies completely to the right of the threshold, signaling that fragtemplates are not accepted by the external system and hence succeed in providing protection. Basically, the limited amount of information in them is not sufficient to maintain the overall HD below the threshold. Note that, even if small differences between the distributions remain, they do not constitute a problem since the curves are to the right of the threshold (i.e., their score is above the acceptance threshold implying a failure in the identity verification). As a matter of fact, the goal of fragmentation is not to make the genuine and impostor distributions identical, but to prevent successful impersonation (i.e., the matching scores produced by fragtemplates should not satisfy the acceptance criterion of the iris recognition system).

In the realistic scenario, the curves see a similar trend. However, as it is visible in Figure 8(e)-(h) even the starting scenario with $f = 4$ enjoys some protection. Again, as the number of fragments grows, the green curve moves to the right, reaching complete protection (green curve area to the right of the threshold) with $f = 10$, implying no fragtemplate succeeds in the matching. When $f = 20$, the genuine fragtemplate and impostor score distributions show a strong overlap. Again, while not perfectly identical, they appear to the right of the threshold corresponding to unsuccessful identity verifications. In all the experiments, we also measured the Genuine Acceptance Rate (GAR), that is, the percentage of identification tests with HD lower than the threshold for which the subject was correctly identified [23]. The GAR values confirm the observations above, with GAR reaching 0 for $f = 20$ in the worst-case scenario and for $f = 10$ and $f = 20$ in the realistic scenario.

6 Related work

Existing solutions for protecting biometric templates mainly rely on cancelable biometrics, biometric cryptosystems, and cryptographic techniques (e.g., [1, 11, 25]). Cancelable biometrics apply intentional and repeatable transformations to biometric data, enabling comparisons to be performed in the transformed domain (e.g., [21, 27, 31, 32, 34]). Although effective, these approaches often lead to a degradation in the recognition accuracy. Biometric cryptosystems leverage iris features to protect or derive cryptographic keys (e.g., [7, 19, 29]). The main problem of these solutions is that the intra-class variability of biometric data complicates the design of robust and reliable cryptographic mechanisms, often requiring error-tolerant mechanisms that may degrade recognition performance.

Cryptographic techniques based on homomorphic encryption and secure multiparty computation aim instead at protecting the matching process and the confidentiality of iris templates. Homomorphic encryption enables the computation of the Hamming distance between encrypted iris templates, with a recognition

accuracy comparable to that of non-protected systems, but at a high computational cost (e.g., [4, 20, 26]). Secure multi-party computation protocols (e.g., [5]) instead leave unchanged the iris template representation and enforce protection at the protocol level, typically supporting a secure one-to-many comparison for large-scale biometric deduplication scenarios. Such solutions, however, do not directly consider identification, which can be supported with a significant computational and communication overheads. Differently from these solutions, our fragmentation-based approach operates in the plaintext domain and considers a more challenging application context.

Data fragmentation has been proposed for protecting the confidentiality of data by splitting them into multiple fragments so that no fragment reveals sensitive information. Fragmentation has been used, in combination with AONT, to split encrypted resources in such a way that absence of a single fragment prevents decryption (e.g., [3]). Hence, proper sharding of the encrypted resource and replication of its fragments provide both confidentiality and resilience to failure in DCS (e.g., [2]). Fragmentation can also be used for protecting confidentiality of data stored at servers under the control of third (possibly untrusted) parties, limiting the adoption of encryption (e.g., [6, 10]). Intuitively, splitting data in different fragments guarantees that only authorized parties can reconstruct sensitive information. These approaches cannot be directly applied to biometric templates since a biometric template is a single, structured piece of information that must support matching operations. Also, matching requires the comparison of correlated components of the template (i.e., aligned positions in the iris code and mask). This requires a careful design of fragmentation in such a way that the matching capability is preserved, while ensuring that each fragment does not enable identification.

7 Conclusions

We investigated how fragmentation can be used to provide protection to biometric templates. Our approach for fragmenting iris biometrics partitions templates in fragments, each of which is mapped onto a fragtemplate that will then be leveraged for performing matching at the fragment level. Each fragtemplate contains all synthetic fragments but a genuine one, which remains however indistinguishable as such from the others. Our experiments show that fragmentation has a negligible impact on identification accuracy and how increasing the number of fragments increases protection. They also confirm that fragmenting at too coarse grain (e.g., four fragments) is insufficient, but in realistic scenarios ten fragments provide complete protection while not affecting accuracy. The paper leaves space for future work including alternative fragmentation strategies, extension to other biometrics, the consideration of alternative protection scenarios, and the combination of fragmentation with lightweight cryptographic safeguards.

Acknowledgements. This work was supported in part by project SERICS (PE00000014) under the MUR NRRP funded by the EU - NGEU.

References

1. Sani M. Abdullahi, Shuifa Sun, Beng Wang, Ning Wei, and Hongxia Wang. Biometric template attacks and recent protection mechanisms: A survey. *Information Fusion*, 103:102144, March 2024.
2. Enrico Bacis, Sabrina De Capitani di Vimercati, Sara Foresti, Stefano Paraboschi, Marco Rosa, and Pierangela Samarati. Securing resources in decentralized cloud storage. *IEEE Transactions on Information Forensics and Security (TIFS)*, 15(1):286–298, December 2020.
3. Enrico Bacis, Sabrina De Capitani di Vimercati, Sara Foresti, Stefano Paraboschi, Marco Rosa, and Pierangela Samarati. Mix&slice for efficient access revocation on outsourced data. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 21(3):1390–1405, May/June 2024.
4. Mauro Barni, Tiziano Bianchi, Dario Catalano, Mario Di Raimondo, Ruggero Donida Labati, Pierluigi Failla, Dario Fiore, Riccardo Lazzeretti, Vincenzo Piuri, Fabio Scotti, and Alessandro Piva. Privacy-preserving fingercode authentication. In *Proc. of the 12th ACM Workshop on Multimedia and Security (MM&Sec)*, Rome, Italy, September 2010.
5. Remco Bloemen, Bryan Gillespie, Daniel Kales, Philipp Sippl, and Roman Walch. Large-scale MPC: Scaling private iris code uniqueness checks to millions of users. Cryptology ePrint Archive, Paper 2024/705, 2024.
6. Valentina Ciriani, Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Combining fragmentation and encryption to protect privacy in data storage. *ACM Transactions on Information and System Security (TISSEC)*, 13(3):22:1–22:33, July 2010.
7. Priyabrata Dash, Fagul Pandey, Monalisa Sarma, and Debasis Samanta. Efficient private key generation from iris data for privacy and security applications. *Journal of Information Security and Applications*, 75:103506, June 2023.
8. John Daugman. The importance of being random: statistical principles of iris recognition. *Pattern Recognition*, 36(2):279–291, February 2003.
9. John Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):21–30, January 2004.
10. Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Giovanni Livraga, Stefano Paraboschi, and Pierangela Samarati. Fragmentation in presence of data dependencies. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 11(6):510–523, November/December 2014.
11. Ruggero Donida Labati, Vincenzo Piuri, and Fabio Scotti. Biometric privacy protection: Guidelines and technologies. In *E-Business and Telecommunications*, pages 3–19, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
12. European Parliament and Council. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance), 2016.
13. European Parliament and Council. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act) (Text with EEA relevance), 2024.

14. Heinz Hofbauer, Fernando Alonso-Fernandez, Peter Wild, Josef Bigun, and Andreas Uhl. A ground truth for iris segmentation. In *Proc. of the 22nd International Conference on Pattern Recognition (ICPR)*, Stockholm, Sweden, August 2014.
15. Heinz Hofbauer, Ehsaneddin Jalilian, and Andreas Uhl. Exploiting superior CNN-based iris segmentation for better recognition accuracy. *Pattern Recognition Letters*, 120:17–23, April 2019.
16. Anil K. Jain, Arun Ross, and Karthik Nandakumar. *Introduction to Biometrics*. Springer, 2011.
17. Hugo Krawczyk, Mihir Bellare, and Ran Canetti. HMAC: Keyed-hashing for message authentication. RFC 2104, February 1997.
18. Ajay Kumar and Arun Passi. Comparison and combination of iris matchers for reliable personal authentication. *Pattern Recognition*, 43(3):1016–1026, March 2010.
19. Youn Joo Lee, Kwanghyuk Bae, Sung Joo Lee, Kang Ryoung Park, and Jaihie Kim. Biometric key binding: Fuzzy vault based on iris images. In Seong-Whan Lee and Stan Z. Li, editors, *Advances in Biometrics*, pages 800–808, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
20. Ying Luo, Sen-ching Samson Cheung, Tommaso Pignata, Riccardo Lazzeretti, and Mauro Barni. An efficient protocol for private iris-code matching by means of garbled circuits. In *Proc. of the 19th IEEE International Conference on Image Processing (ICIP)*, Orlando, FL, USA, September-October 2012.
21. Manisha and Nitin Kumar. Cancelable biometrics: A comprehensive survey. *Artificial Intelligence Review*, 53(5):3403–3446, 2020.
22. Libor Masek and Peter Kovesi. MATLAB source code for a biometric identification system based on iris patterns. The School of Computer Science and Software Engineering, The University of Western Australia, 2003.
23. Evangelia Micheli-Tzanakou, Konstantinos N. Plataniotis, and Ruggero Donida Labati. Biometrics: Terms and definitions. In Sushil Jajodia, Pierangela Samarati, and Moti Yung, editors, *Encyclopedia of Cryptography, Security and Privacy*, pages 1–5. Springer Berlin Heidelberg, Berlin, Heidelberg, 2019.
24. Mahsa Mitcheff, Patrick Tinsley, and Adam Czajka. Privacy-safe iris presentation attack detection. In *Proc. of the IEEE International Joint Conference on Biometrics (IJCB)*, Buffalo, NY, USA, September 2024.
25. Bill Moriarty, Chun-Fu Chen, Shaohan Hu, Sean Moran, Marco Pistoia, Vincenzo Piuri, and Pierangela Samarati. Utility-preserving biometric information anonymization. In *Proc. of the 27th European Symposium on Research in Computer Security (ESORICS)*, Copenhagen, Denmark, September 2022.
26. Shenghui Peng, Peiheng Jia, Jinbo Xiong, Liehuang Zhu, and Ximeng Liu. Privacy preservation fully homomorphic encryption for cloud-assisted biometric identification with multi-key. In *Proc. of the IEEE Global Communications Conference (GLOBECOM)*, Cape Town, South Africa, December 2024.
27. Jaishanker K. Pillai, Vishal M. Patel, Rama Chellappa, and Nalini K. Ratha. Sectored random projections for cancelable iris biometrics. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, TX, USA, March 2010.
28. George Quinn, Patrick Grother, and Elham Tabassi. *Standard Iris Storage Formats*, pages 89–100. Springer London, London, 2016.
29. Christian Rathgeb, Andreas Uhl, and Peter Wild. *Iris Biometric Cryptosystems*, pages 203–222. Springer New York, New York, NY, 2013.
30. Christian Rathgeb, Andreas Uhl, Peter Wild, and Heinz Hofbauer. Design decisions for an iris recognition SDK. In Kevin Bowyer and Mark J. Burge, editors, *Hand-*

book of Iris Recognition, Advances in Computer Vision and Pattern Recognition. Springer, second edition, 2016.
31. Chong Siew Chin, Andrew Teoh Beng Jin, and David Ngo Chek Ling. High security iris verification system based on random secret integration. *Computer Vision and Image Understanding*, 102(2):169–177, May 2006.
32. Wiraj Udara Wickramaarachchi, Dongdong Zhao, Junwei Zhou, and Jianwen Xiang. An effective iris biometric privacy protection scheme with renewability. *Journal of Information Security and Applications*, 80:103684, February 2024.
33. Shivangi Yadav and Arun Ross. iWarpGAN: Disentangling identity and style to generate synthetic iris images. In *Proc. of the IEEE International Joint Conference on Biometrics (IJCB)*, Ljubljana, Slovenia, September 2023.
34. Qianrong Zheng, Jianwen Xiang, Changtian Song, Rivalino Matias, Rui Hao, Songsong Liao, Xuemin Zhang, Meng Zhao, and Dongdong Zhao. Cancelable iris template based on slicing. *Computers & Security*, 152:104381, May 2025.