Exploit the Leak: Understanding Risks in Biometric Matchers

Dorine Chagnon[1], Axel Durbet[1][a], Paul-Marie Grollemund[2][b]
and Kevin Thiry-Atighehchi[1,*][c]

[1] *University Clermont Auvergne, LIMOS (UMR 6158 CNRS), Clermont-Ferrand, France*

[2] *University Clermont Auvergne, LMBP (UMR 6620 CNRS), Clermont-Ferrand, France*

[*] *kevin.atighehchi@uca.fr*

Keywords: Privacy-Preserving Distance, Hamming Distance, Information Leakage, Biometric Security.

Abstract: In a biometric authentication or identification system, the matcher compares a stored and a fresh template to determine whether there is a match. This assessment is based on both a similarity score and a predefined threshold. For better compliance with privacy legislation, the matcher can be built upon a privacy-preserving distance. Beyond the binary output ('*yes*' or '*no*'), most schemes may perform more precise computations, *e.g.*, the value of the distance. Such precise information is prone to leakage even when not returned by the system. This can occur due to a malware infection or the use of a weakly privacy-preserving distance, exemplified by side channel attacks or partially obfuscated designs. This paper provides an analysis of information leakage during distance evaluation. We provide a catalog of information leakage scenarios with their impacts on data privacy. Each scenario gives rise to unique attacks with impacts quantified in terms of computational costs, thereby providing a better understanding of the security level.

1 INTRODUCTION

Biometric authentication protocols involve the comparison of a fresh biometric template with the reference template. This comparison computes the distance between the newly acquired data and the stored template. If this distance is below a given threshold, access is granted; otherwise, it is denied. Hamming distance is a widely used metric in biometric applications *e.g.*, biohashing (Patel et al., 2015; Bernal-Romero et al., 2023), iriscode (Daugman, 2009; Dehkordi and Abu-Bakar, 2015; Daugman, 2015), face recognition (Yang and Wang, 2007; He et al., 2015), gait recognition (Tran et al., 2017), keystroke (Rahman et al., 2021), ear authentication (Wang et al., 2021) and palm-vein recognition (Cho et al., 2021). Computing this distance may inadvertently leak information that adversaries might exploit to reconstruct the stored template. These vulnerabilities may arise from implementation errors, inherent flaws, and server-level attacks such as malware (Sharma et al., 2023), which can compromise system-wide security. Furthermore, Aydin and Aysu (Aydin and Aysu, 2024) and Hashemi *et al.* (Hashemi et al., 2024) have highlighted an increasing prevalence of side-channel attacks. Side-channel techniques, including timing, differential power analysis, cache-based, electromagnetic, acoustic, and thermal attacks, exploit various operational artifacts to extract sensitive information (Sharma et al., 2023). One of the concerns is the partial or total leakage of distance computation information. Such information leakage poses significant security and privacy risks, especially in sensitive applications like privacy-preserving applications (*e.g.*, biometric recognition systems). In this paper, we focus on the following attacks:

- *Offline exhaustive search attacks* refer to scenarios for which a leaked yet obfuscated database is available for an attacker. The attacker employs the public transformation to verify a candidate vector. This verification may give additional information beyond the minimal information leakage (*'yes'* or *'no'*), for example via side-channel attacks.
- *Online exhaustive search attacks* correspond to attacks for which an attacker must interact with the biometric system to infer information about the targeted vector. Then, the attacker needs to force the system to leak additional information beyond the minimal information leakage (*'yes'* or *'no'*),

[a] https://orcid.org/0000-0002-4420-1934

[b] https://orcid.org/0000-0002-1273-1658

[c] https://orcid.org/0000-0003-0042-8771

Chagnon, D., Durbet, A., Grollemund, P.-M. and Thiry-Atighehchi, K.
Exploit the Leak: Understanding Risks in Biometric Matchers.
DOI: 10.5220/0013250600003899
In *Proceedings of the 11th International Conference on Information Systems Security and Privacy (ICISSP 2025) - Volume 2*, pages 353-362
ISBN: 978-989-758-735-1; ISSN: 2184-4356

for example via a malware infection.

Related Works. To the best of our knowledge, two papers investigate information leakage of biometric systems using privacy-preserving distance. Pagnin *et al.* (Pagnin et al., 2014) shows that the output of a privacy-preserving distance can be exploited to infer the hidden input. This type of attack is considered the most devastating for such systems, as evidenced by Simoens *et al.* (Simoens et al., 2012). The work of Pagnin *et al.* takes place in the minimal leakage scenario, wherein only the binary output of the biometric system is given to the attacker. The authors present the *Center Search Attack*, designed to recover the hidden enrolled input for any 'valid' biometric template in $\mathbb{Z}_q^n$, where 'valid' refers to inputs within a ball centered at the enrolled template and with a radius equal to the decision threshold t. To efficiently locate a valid input, the authors also examine the exhaustive search attack, particularly its application on binary templates ($q = 2$). They suggest implementing a *sampling without replacement* strategy using their *Tree algorithm* to streamline the identification of a suitable input for the Center Search Attack. This efficient identification of a proper input requires a number of authentication attempts that is exponential in the space dimension n minus the threshold t. While their work focuses on the minimal leakage scenario, our analysis includes the consideration of multiple additional information leaks that may arise during the matching operation.

Contributions. We analyze the impact of potential information leakage in distance evaluations. Our contributions detail various leakage scenarios, their corresponding generic attacks, and the computational costs involved:

- We revisit the exhaustive search attack in the scenario of a minimal (one-bit) information leakage, correcting a previous result (see (Pagnin et al., 2014)) about the costs of optimal and near-optimal strategies and include additional information on cases that are not well-detailed in the literature.
- We introduce new attack strategies by malicious clients that exploit various levels of non-minimal information leaks from the system. Our complexity results, which detail the cost of these attacks, apply to both *offline exhaustive search attacks* that leverage a leaked (yet obfuscated) database and *online exhaustive search attacks* involving direct interactions with the server.
- We investigate a novel attack, named accumulation attack, where an *honest-but-curious* server accumulates knowledge during client authentication. This type of attack occurs when there is a minor, yet non-negligible, amount of information leakage.

The complexities of the attacks, relying on different scenarios, are summarized in Table 1.

Outline. Section 2 introduces notations and terminologies and classifies the different types of information leakages. Section 3 begins by revisiting the exhaustive search attack in the minimal (one-bit) information leakage scenario, including a correction of a previously cited result concerning the costs of optimal and near-optimal strategies. It then introduces new strategies for attacks by malicious clients capturing various other types of information leakages, covering both offline and online exhaustive search attacks, with an emphasis on their computational costs. The section concludes by examining accumulation attacks performed by an "honest-but-curious" server during client authentication, detailing the computational cost involved. Section 4 provides a discussion of the presented results.

2 PRELIMINARIES

This section introduces the notations as well as the attacker model and, a list of the considered information leakage scenarios.

2.1 Notations and Attacker Models

Let $\mathbb{Z}_q^n = \{0,\ldots,q-1\}^n$ be a metric space equipped with the Hamming distance d and $\varepsilon \in \mathbb{N}$ a threshold. The Hamming distance is defined by

$$d(x,y) = |\{i \in \{1,\ldots n,\} | x_i \neq y_i\}|$$

for two vectors $x = (x_1,\ldots,x_n)$ and $y = (y_1,\ldots,y_n)$ in $\mathbb{Z}_q^n$. Let $\mathtt{Match}_{x,\varepsilon}$ denote the oracle modeling the interaction between the biometric system using a privacy-preserving distance and the attacker. $\mathtt{Match}_{x,\varepsilon}$ receives the template selected by the attacker and compares it with the previously enrolled and stored template. If the distance is below the threshold ε, the oracle returns 1 and 0 otherwise. In a more formal way, $\mathtt{Match}_{x,\varepsilon}$ is a function defined as:

$$\begin{array}{rcl} \mathtt{Match}_{x,\varepsilon} : \mathbb{Z}_q^n & \longrightarrow & \{0,1\} \\ y & \longmapsto & \begin{cases} 1 \text{ if } d(x,y) \leq \varepsilon. \\ 0 \text{ otherwise.} \end{cases} \end{array}$$

A privacy-preserving distance may leak additional information beyond its binary output. Under the specifications of each scenario, the oracle may display this

Table 1: Summary of all leakage exploits and their complexities with α such that the occurrence of the rarest error is $n^{-\alpha}$ with $\alpha \in \mathbb{R}_{\geq 1}$. The Distance-to-Threshold comparison determines if the leak occurs when $d(x,y) \leq \varepsilon$ (below) or when there is no distance requirement between x and y (both). For all the complexities, x and y are in $\mathbb{Z}_q^n$ with $q \geq 2$ except for the minimal leakage where x and y are in $\mathbb{Z}_2^n$. The provided complexities represent worst-case scenarios, except for the accumulation attack where the result is the expectation.

Distance-to-Threshold comparison	Leakage	Complexity type	Complexity in Big-Oh	Theorem
Below	Distance	Exponential	$q^{n-\varepsilon} + q\varepsilon$	3.2
	Positions	Exponential	$q^{n-\varepsilon} + q$	3.3
	Positions and values	Exponential	$q^{n-\varepsilon}$	3.4
	Positions and values (accumulation)	Linearithmic/Polynomial	$n^{\alpha} \log n$	3.9
Both	Minimal[1]	Exponential	$q^{n-\varepsilon} + n(q-1) + 2\varepsilon$	3.5
	Distance	Linear	nq	3.6
	Positions	Constant	q	3.7
	Positions and values	Constant	1	3.8

[1] Note that the Big-Oh complexity of the optimal exhaustive search strategy, in the worst-case, is the same as the naive strategy as the minimum of $h(\cdot)$ is 0.

additional information. The objective of the attacker is to find the hidden template x exploiting the oracle outputs. In the context of a biometric system, the objective of the attacker may be relaxed to simply find y that is close to x with respect to d and ε.

2.2 Typology of Information Leakage

In the context of a biometric system, a critical vulnerability arises when information is intercepted between the matcher and the decision module, as illustrated in Figure 1 (point 8). This figure, inspired by Ratha *et al.* (Ratha et al., 2001), provides an overview of the attack points in biometric systems while introducing both the decision module and two additional attack points. Except for the accumulation attack, the attacker exploits points 4 and 8 in all discussed scenarios. Point 4 allows the submission of a chosen template, while point 8 grants access to additional information beyond the binary output. The accumulation attack only necessitates control over the point 8. For detailed insights into the remaining attack points, readers are referred to Ratha *et al.* (Ratha et al., 2001). There are three main categories of information leakage: Below the threshold; Above the threshold; Both below and above the threshold.

In each of these categories, several sub-settings can be identified. The first one corresponds to the absence of any leakage, resulting in $\texttt{Match}_{x,\varepsilon}$ yielding only the binary output. Then, the following information leakages are examined:

- The distance.
- The positions of the errors.
- Both the error positions and values.
- Both the distance and the positions of the errors.
- Both the distance and the positions and their corresponding erroneous values.

It is not relevant to consider that additional information is leaked only above the threshold, as no scheme has such behavior. As a consequence, solely scenarios ‘below the threshold’ and ‘below and above the threshold’ are examined. The Hamming distance is a measure of the number of differing coordinates between two templates. Therefore, knowledge of the erroneous coordinates implies knowledge of the distance itself. Hence, we do not consider all possible scenarios.

3 EXPLOITING THE LEAKAGE

This section provides a comprehensive analysis of the attacks that can be performed in each leakage scenario, along with an evaluation of their complexity.

3.1 Active Attacks

This section focuses on active attacks, *i.e.*, attacks where the attacker submits templates to the oracle $\texttt{Match}_{x,\varepsilon}$.

3.1.1 Attack Complexity for the Minimal (One-Bit) Leakage

In this section, the attacker aims to find a template that lies in the ball of center x (the target template) and radius ε (the threshold). To identify such a point, several methods are available, each with its own set of advantages and disadvantages.

Brute Force. The objective of this attack is to exhaustively test all possible templates until the oracle $\texttt{Match}_{x,\varepsilon}$ yields 1. In the worst case, we test every template, which results in the examination of q^n vectors. To obtain this result, we ignore the ε acceptance

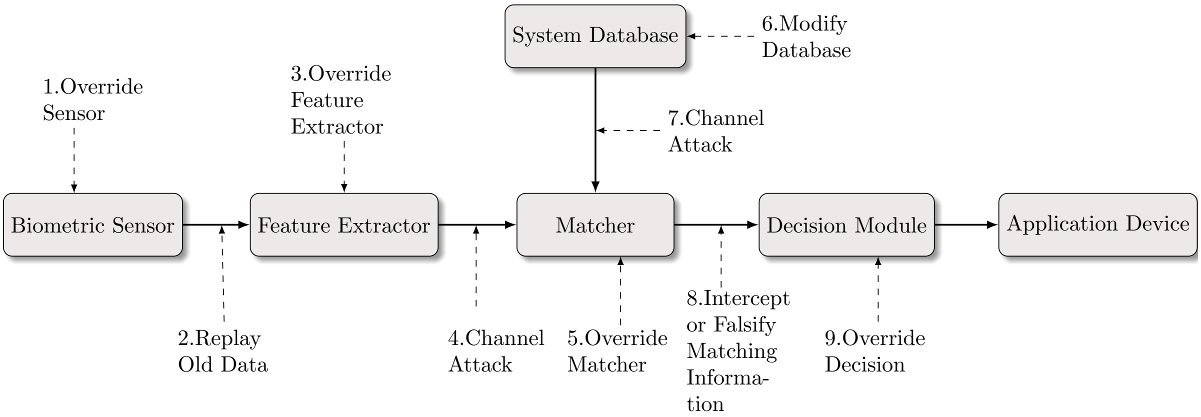

Figure 1: Attack points in a generic biometric recognition system.

threshold. On the other hand, if we consider that only $n-\varepsilon$ exact coordinates are needed to be accepted by the system, complexity decreases to $q^{n-\varepsilon}$ tests. Since the attacker specifically targets $n-\varepsilon$ coordinates (the attacker arbitrarily chooses ε coordinates that do not change), and aims for a perfect match for the $n-\varepsilon$ remaining coordinates yielding the result.

Random Sampling. The attacker randomly chooses a template in $\mathbb{Z}_q^n$ and tests it by querying the oracle $\mathtt{Match}_{x,\varepsilon}$. The precise complexity of this strategy has not been assessed in the literature. The worst case for the attacker occurs when the templates are uniformly distributed in $\mathbb{Z}_q^n$. The probability that a template submitted to $\mathtt{Match}_{x,\varepsilon}$ yields 1 is $\rho = \frac{|B_{q,\varepsilon}(x)|}{q^n}$. According to this naive strategy, we can assume that the tests are independent and that each is modeled as a Bernoulli experiment with a success probability of ρ. The number of tries needed to obtain the first success follows a geometric distribution. Hence, the expected number of tries for an attacker to get accepted by the system is p^{-1}. First, recall that the cardinal of $B_{q,\varepsilon}(x)$ is

$$|B_{q,\varepsilon}(x)| = \sum_{i=0}^{\varepsilon} \binom{n}{i} (q-1)^i,$$

and that the q-ary entropy is $h_q(x) = x\log_q(q-1) - x\log_q x - (1-x)\log_q(1-x)$. Then, using the Stirling approximation (see (Timothée and Ramanna, 2016; Thomas and Joy, 2006)), the expected number of tries for an attacker is

$$\begin{aligned} \rho^{-1} &= \frac{q^n}{|B_{q,\varepsilon}(x)|} = \frac{q^n}{\sum\limits_{i=0}^{\varepsilon} \binom{n}{i}(q-1)^i} \\ &\leq \frac{q^n}{q^{nh_q(\varepsilon/n)+o(n)}} = q^{n(1-h_q(\varepsilon/n))+o(n)} \end{aligned}$$

if $\frac{\varepsilon}{n} \leq 1 - \frac{1}{q}$ holds, and if n is large enough.

Random Sampling Without Point Replacement. As the random sampling, the attacker randomly chooses a template in the set $S \subseteq \mathbb{Z}_q^n$. At each step, if $\mathtt{Match}_{x,\varepsilon}$ returns 0, the tested vector b is removed from the set S. The probability of success does not remain constant throughout the experiment, unlike in the previous case. Consequently, the experiment follows a hypergeometric distribution. This game is equivalent to having an urn with q^n object where $|B_{q,\varepsilon}(x)|$ are considered 'good'. Then, according to Ahlgren (Ahlgren, 2014) the expected number of queries to $\mathtt{Match}_{x,\varepsilon}$ before success is given by

$$\frac{q^n+1}{|B_{q,\varepsilon}(x)|+1} \approx \rho^{-1}.$$

This attack has a slightly better performance compared to the previous one, although it is accompanied by an exponential memory cost that reduces its efficiency, making this version less interesting than the previous one.

Remark 3.1.1. *In the case of random sampling, if the value of n is large, it is preferable to select a draw with replacement to save memory while maintaining a high degree of performance. Indeed, the probability of drawing a vector that has already been selected is relatively small if n is sufficiently large.*

Tree Search. This algorithm was proposed by Pagnin *et al.* (Pagnin et al., 2014). The underlying idea is to construct a tree of depth n such that each point of the space is considered to be a leaf. The tree structure is utilized to establish relative relations among the points of $\mathbb{Z}_q^n$ and to guarantee that after each unsuccessful trial, non-overlapping portions of the space $\mathbb{Z}_q^n$ can be removed. Specifically, if a point $p \in \mathbb{Z}_q^n$ does not satisfy the authentication, the algorithm removes not only the tested point p from the set of potential centers but also its sibling relatives generated by the common ancestor ε (*i.e.*, the subtree of height ε covering these siblings is removed). At each attempt, the attacker can remove approximately q^{ε} templates from the research space (for more details, please refer to (Pagnin et al., 2014)). The running time of the attack is the cost of exploring a q-ary tree of order $n-\varepsilon$.

Remark 3.1.2. *It should be noted that, as intended, the cost of all the presented attacks is exponential.*

Optimal Solution. The optimal solution is to solve the set-covering problem (Korte et al., 2011) using balls of radius ε. The main idea is to cover the space with the smallest number of balls of radius ε to partition the space. The objective is to remove an entire ball of radius ε if the query fails. This is an instance of the set covering problems. Pagnin *et al.* (Pagnin et al., 2014) claimed that the number of points that the adversary needs to query is only a factor of $O(\varepsilon \ln(n+1))$ more than the optimal cover. However, the result is imprecise, as detailed below in this remark, mainly because the optimal cover is not given.

Theorem 3.1. *Given ε a threshold, $x \in \mathbb{Z}_q^n$ a vector, and* $\texttt{Match}_{x,\varepsilon}$*, an attacker with the optimal strategy can retrieve x in* $q^{n(1-h_q(\varepsilon/n))+o(n)}$ *queries to* $\texttt{Match}_{x,\varepsilon}$.

Proof. The strategy between a bounded and an unbounded adversary may differ as detailed in the following:

- **Unbounded Adversary:** The adversary solves the NP-hard set covering problem (Korte et al., 2011) to find the optimal covering of $\mathbb{Z}_q^n$ using balls of radius ε. The adversary exhaustively searches x using at most $q^{n(1-h_q(\varepsilon/n))+o(n)}$ queries to $\texttt{Match}_{x,\varepsilon}$. The number of vectors involved in a given optimal cover is $\frac{q^n}{|B_{q,\varepsilon}(x)|}$, which can be asymptotically approximated as detailed in what follows. Then, using bounds on the binomial coefficient (see (Thomas and Joy, 2006; Timothée and Ramanna, 2016)), the result follows if $\frac{\varepsilon}{n} \leq 1 - \frac{1}{q}$ holds and if n is large enough.
- **Bounded Adversary:** The adversary may use a greedy algorithm to find a non-optimal covering containing $\frac{q^n H(n)}{|B_{q,\varepsilon}|}$ vectors (Chvatal, 1979) with $H(n) = \sum_{i=1}^{n} i^{-1}$ the n-th harmonic number. The adversary then finds a solution with an exhaustive search in at most $\frac{q^n H(n)}{|B_{q,\varepsilon}|}$ queries. To provide a more intuitive value, notice that $\frac{q^n H(n)}{|B_{q,\varepsilon}|}$ can be bounded up by $\frac{q^n(\ln(n)+1)}{|B_{q,\varepsilon}|}$. As in the unbounded case, using the q-ary entropy and Stirling's approximation, this non-optimal covering leads the attacker to make at most $q^{n(1-h_q(\varepsilon/n))+o(n)}$ queries, as $\log_q(\ln(n)+1) = o(n)$.

Table 2: Expected number of calls to `oracle` for the exhaustive search method Random Sampling with Replacement (RSR). Examples with real biometric systems with $q=2$.

System	n	ε	RSR ($\log_2$)
IrisCode (Daugman, 2009)	2,048	738	121.37
IrisCode (Daugman, 2009)	2,048	656	199.94
IrisCode (Daugman, 2009)	2,048	574	300.24
FingerCode (Harikrishnan et al., 2024)	80	30	5.92
BioHashing (Belguechi et al., 2013)	180	60	17.74
BioEncoding (Ouda et al., 2010)	350	87	70.62
BioEncoding (Ouda et al., 2010)	350	105	45.18

Then, in both cases, the number of queries is $q^{n(1-h_q(\varepsilon/n))+o(n)}$ and the result follows. ■

Remark 3.1.3. *The time required to configure the greedy algorithm is exponential, rendering the aforementioned attack impractical. Moreover, even if an attacker computes the optimal covering, it still needs to query an exponential number of times the* $\texttt{Match}_{x,\varepsilon}$ *to find a point close to x.*

It is also interesting to note that the expected time for an attacker to be accepted by the system using the random sampling with and without replacement method is equivalent to the worst case using the optimal method.

Example of Expectations for the Random Sampling. To illustrate the influence of the threshold and the choice of q on exhaustive search, we calculate the precise expectation of the number of attempts required for an attacker to successfully impersonate the user in different settings using the random sampling method. The results are presented in Table 2. Experimental results show that to increase the security against exhaustive search, it is more interesting to increase q than to decrease ε.

3.1.2 Attack Complexities for Leakage Below the Threshold

Leakage below the threshold is considered in this section. Given the hidden target x, querying y such that $d(x,y) \leq \varepsilon$ to the oracle $\texttt{Match}_{x,\varepsilon}$ provides information beyond the binary output.

Leakage of the Distance. The first case occurs when the distance is given to the attacker as extra information.

Theorem 3.2. *Given ε a threshold, $x \in \mathbb{Z}_q^n$ a vector, and $\mathtt{Match}_{x,\varepsilon}$ leaks the distance below the threshold, an attacker can retrieve x in the worst case in $O(q^{n-\varepsilon} + q\varepsilon)$ queries to $\mathtt{Match}_{x,\varepsilon}$.*

Proof. The system, using the Hamming distance, requires a minimum of $n - \varepsilon$ accurate coordinates to output 0. Since the attacker specifically targets $n - \varepsilon$ coordinates (the attacker arbitrarily chooses ε coordinates that do not change), an exhaustive search attack is performed in at most $q^{n-\varepsilon}$ steps to get accepted by the system. Then, a hill-climbing attack runs on the remaining ε coordinates to minimize the distance at each step. Coordinate by coordinate, the attacker obtains the right value if the distance decreases. Since there are q different values to test on ε coordinates, determining the correct ones requires a maximum of $(q-1)\varepsilon$ steps. Then, the overall complexity is $O(q^{n-\varepsilon} + q\varepsilon)$. ■

Leakage of the Positions. The positions of the errors are the extra information given to the attacker, while their values remain secret.

Theorem 3.3. *Given ε a threshold, $x \in \mathbb{Z}_q^n$ a vector, and $\mathtt{Match}_{x,\varepsilon}$ leaks the positions of the errors below the threshold, an attacker can retrieve x in the worst case in $O(q^{n-\varepsilon} + q)$ queries to $\mathtt{Match}_{x,\varepsilon}$.*

Proof. As the leakage occurs solely below the threshold, the first step is to find a vector $y \in \mathbb{Z}_q^n$ such that $d(x,y) \leq \varepsilon$. To identify such a vector, the attacker performs an exhaustive search attack in $q^{n-\varepsilon}$ steps, as previously shown. Since ε coordinates remain unknown, and each coordinate ranges from 0 to $q-1$, every possibility must be examined. By testing all possibilities simultaneously – for instance, testing all coordinates at 0, then all coordinates at 1, and so forth up to $q-2$ while retaining the correct values – the original vector can be identified in no more than $q-1$ queries (refer to the example illustrated in Figure 2). Therefore, the complexity of the attack for recovering x is $O(q^{n-\varepsilon} + q)$. ■

Figure 2 gives a representation of the attack described above in the case $\mathbb{Z}_4^5$ and the hidden vector or the missing coordinates is $(0,1,3,2,2)$. Note that the actual complexity is $q-1$ since the final exchange is unnecessary, as the coordinates at $q-1$ become known after $q-1$ queries by inference.

Leakage of the Positions and the Values. When a vector below the threshold is given to the oracle $\mathtt{Match}_{x,\varepsilon}$, the attacker gets information about both error positions and their values. This is similar to an error-correction mechanism designed to correct errors below a given threshold. Note that in the binary case, this scenario is the same as the previous one, hence the only considered case is $q > 2$.

Theorem 3.4. *Given ε a threshold, $x \in \mathbb{Z}_q^n$ a vector, and $\mathtt{Match}_{x,\varepsilon}$ leaks the positions and the values of the errors below the threshold, an attacker can retrieve x in $O(q^{n-\varepsilon})$ queries to $\mathtt{Match}_{x,\varepsilon}$.*

Proof. First, an exhaustive search is performed to find a vector y for which the distance is below the threshold, for a cost of $O(q^{n-\varepsilon})$. Then, given the error positions and the corresponding error values, y yields immediately the recovery of x. In the end, the complexity of the attack is $O(q^{n-\varepsilon})$. ■

3.1.3 Leakage Below and Above the Threshold

The second scenario is considered in this section, which involves a leakage independent of the threshold. In other words, when a hidden vector x is targeted, the queried vector y to the oracle $\mathtt{Match}_{x,\varepsilon}$ results in the leak of additional information.

Minimal Leakage (a Single Bit of Information Leakage). The basic usage of the system is characterized by the minimal leakage scenario, where the binary output itself is considered a necessary leakage. This minimal leakage is indispensable for the system's work and is consistent across these scenarios as the system always responds. Remark that if the server does not answer above the threshold, the non-answer gives the attacker the wanted information.

Theorem 3.5. *Given ε a threshold, $x \in \mathbb{Z}_q^n$ a vector, and $\mathtt{Match}_{x,\varepsilon}$ that does not leak any extra information, an attacker can retrieve x in $O(q^{n-\varepsilon} + n(q-1) + 2\varepsilon)$ queries to $\mathtt{Match}_{x,\varepsilon}$.*

Proof. As in the previous cases, the attacker seeks a vector y below the threshold. Such a vector is found by exhaustive search in $q^{n-\varepsilon}$ steps. Then, the attacker performs the center search attack (Pagnin et al., 2014) (generalized to $\mathbb{Z}_q^n$) to retrieve the original data in at most $n(q-1) + 2\varepsilon$ queries. Indeed, the generalization does not change the cost of the edge detection but changes the cost of the center search from n to $n(q-1)$. The complexity of the attack to find x is $O(2^{n-\varepsilon} + n + 2\varepsilon)$. ■

Leakage of the Distance. In this case, $d(x,y)$ the distance between $y \in \mathbb{Z}_q^n$ the fresh template and $x \in \mathbb{Z}_q^n$ the old template is leaked to the attacker regardless of the threshold.

$$\begin{array}{rccccc}
\underline{\text{Queries:}} & (\boxed{0}, & \underset{\times}{0}, & \underset{\times}{0}, & \underset{\times}{0}, & \underset{\times}{0}) \\
 & (\underset{\times}{1}, & \boxed{1}, & \underset{\times}{1}, & \underset{\times}{1}, & \underset{\times}{1}) \\
 & (\underset{\times}{2}, & \underset{\times}{2}, & \underset{\times}{2}, & \boxed{2}, & \boxed{2}) \\
 & (\underset{\times}{3}, & \underset{\times}{3}, & \boxed{3}, & \underset{\times}{3}, & \underset{\times}{3}) \\
\underline{\text{Solution:}} & (0, & 1, & 3, & 2, & 2)
\end{array}$$

Figure 2: Exploiting the error position leaked in the case $\mathbb{Z}_4^5$ and the hidden vector or missing coordinates is $(0,1,3,2,2)$.

Theorem 3.6. *Given* ε *a threshold,* $x \in \mathbb{Z}_q^n$ *a vector, and* $\mathtt{Match}_{x,\varepsilon}$ *leaks the distance, an attacker can retrieve* x *in* $O(nq)$ *queries to* $\mathtt{Match}_{x,\varepsilon}$.

Proof. As the attacker has access to the distance, it is possible to perform a hill-climbing attack, trying to minimize the distance at each step. The strategy is to find the vector y, coordinate by coordinate. As each coordinate has q possible values and there are n coordinates, this is done in $O(nq)$ steps. ■

Leakage of the Positions. The extra information given to the attacker is the positions of the errors.

Theorem 3.7. *Given* ε *a threshold,* $x \in \mathbb{Z}_q^n$ *a vector, and* $\mathtt{Match}_{x,\varepsilon}$ *leaks the positions of the errors, an attacker can retrieve* x *in* $O(q)$ *queries to* $\mathtt{Match}_{x,\varepsilon}$.

Proof. She tries the vector $(0,\ldots,0)$, $(1,\ldots,1)$ up to, $(q-1,\ldots,q-1)$ and keep for each coordinate the right value (see Figure 2). Hence, the complexity of the attack to recover x is $O(q)$. ■

Leakage of the Positions and the Values. In this last case, the positions of the errors and corresponding values are leaked. Unlike the scenario of leakage below the threshold, such a leak provides an error-correcting code mechanism that operates irrespective of any distance and threshold.

Theorem 3.8. *Given* ε *a threshold,* $x \in \mathbb{Z}_q^n$ *a vector, and* $\mathtt{Match}_{x,\varepsilon}$ *leaks the positions of the errors and their values, an attacker can retrieve* x *in* $O(1)$ *queries to* $\mathtt{Match}_{x,\varepsilon}$.

Proof. The submission of any vector gives the position of each error, and how to correct them, yielding a complexity in $O(1)$. ■

Example of the Worst Case for Active Attacks Depending on the Leakage. To illustrate the influence of the leakage type on the attack complexity, we compute the number of attempts (in the worst case) required for an attacker to successfully impersonate the user in different settings. The results are presented in Table 3 and Table 4.

3.2 Accumulation Attack: A Passive Attack

During the client authentications, the attacker passively gathers information by observing errors leaked by the server. More specifically, the server leaks a list of positions and errors computed over the integers (*i.e.*, $x_i - y_i$) made by a genuine client during each authentication. Such information gathered during one successful authentication attempt is called an observation. The attacker aims to partially or fully reconstruct x by exploiting these observations.

In the binary case (*i.e.*, $q = 2$), the errors precisely yield the bits. If $x_i - y_i = 1$ then $x_i = 1$, and if $x_i - y_i = -1$ then $x_i = 0$. This attack is related to the Coupon Collector's problem (Ferrante and Saltalamacchia, 2014), which involves determining the expected number of rounds required to collect a complete set of distinct coupons, with one coupon obtained at each round, and each coupon acquired with equal probability.

Example 3.2.1. *Suppose a setting with a metric space* $\mathbb{Z}_2^n$ *equipped with the Hamming distance. A client seeks to authenticate to an honest-but-curious server that uses a scheme leaking* $d(x,y)$ *and the corresponding errors if* $d(x,y) \leq \varepsilon$. *As the client is legitimate, i.e.,* $d(x,y) \leq \varepsilon$ *with a high probability, the attacker recovers the values of at most* ε *erroneous bits. The attacker needs to collect all the bits of the client, turning this problem into a Coupon Collector problem. For example, let assume* $x = (0,0,1,1,0,1,0)$, $\varepsilon = 3$. *The attacker sets* $z = (?,?,?,?,?,?,?)$. *Session 1: The client authenticates with* $y = (1,1,0,1,0,1,0)$. *In this case,* $d(x,y) = 3 \leq \varepsilon$. *The values of the erroneous bits of the client are obtained, yielding* $z = (0,0,1,?,?,?,?)$. *Session 2: the client authenticates with* $y = (0,0,0,0,1,1,0)$. *In this case,* $d(x,y) = 3 \leq \varepsilon$, *and the attacker obtains the value of the erroneous bits of the client and updates* $z = (0,0,1,1,0,?,?)$. *At this point, replacing the unknown values with random bits gives a vector that lies inside the acceptance ball as the number of unknown coordinates is smaller than the threshold* ε.

In biometrics, some errors happen more frequently than others. In this setup, the Weighted Coupon Collector's Problem must be considered. Each coupon (*i.e.*, each error) has a probability p_i to occur. Suppose that $p_1 \leq p_2 \leq \cdots \leq p_n$ and $\sum_{i=1}^{n} p_i \leq 1$ then, according to Berenbrink and Sauerwald (Berenbrink and Sauerwald, 2009) (Lemma 3.2), the expected number of round E is such that:

$$\frac{1}{p_1} \leq E \leq \frac{H(n)}{p_1} \tag{1}$$

Table 3: Number of calls to `oracle` depending on the leakage type (worst case analysis). Examples with real biometric systems (for the leakage below the threshold) with $q = 2$.

System	n	ε	Complexity (log_2)		
			Distance	Position	Distance and Position
IrisCode (Daugman, 2009)	2,048	738	1,310	1,310	1,310
FingerCode (Harikrishnan et al., 2024)	80	30	50	50	50
BioHashing (Belguechi et al., 2013)	180	60	120	120	120
BioEncoding (Ouda et al., 2010)	350	87	263	263	263

Table 4: Number of calls to `oracle` depending on the leakage type (worst case analysis). Examples with real biometric systems (for the leakage both above and below the threshold) with $q = 2$.

System	n	ε	Complexity (log_2)		
			Distance	Position	Distance and Position
IrisCode (Daugman, 2009)	2,048	738	12.00	1	0
FingerCode (Harikrishnan et al., 2024)	80	30	7.32	1	0
BioHashing (Belguechi et al., 2013)	180	60	8.49	1	0
BioEncoding (Ouda et al., 2010)	350	87	9.45	1	0

with $H(n)$ the n-th harmonic number. The upper bound on $H(n)$ is $1 + \log n$, which yields the expected number of rounds required to complete the collection:

$$\frac{1}{p_1} \leq E \leq \frac{\ln(n)+1}{p_1}. \tag{2}$$

However, while in the original problem one coupon is obtained at each round, the number of errors made by a client during an authentication session is variable, *i.e.*, between 1 and ε. In this case, the expected number of rounds required before all the errors have been observed is smaller than in the case where only one error occurs at each round. Consequently, the expected number of rounds required to collect all the errors is still in $O(\log n/p_1)$.

Theorem 3.9. *Given ε a threshold, $x \in \mathbb{Z}_2^n$ a vector, $\texttt{Match}_{x,\varepsilon}$ leaks the positions of the errors and their values below the threshold, and assuming that the rarest coupon is obtained with probability $p_1 = n^{-\alpha}$ with $\alpha \in \mathbb{R}_{\geq 1}$ an attacker can retrieve x in $O(n^\alpha \log n)$.*

Proof. According to the Weighted Coupon Collector's problem and assuming that the rarest coupon is obtained with probability $p_1 = n^{-\alpha}$ with $\alpha \in \mathbb{R}_{\geq 1}$, the vector x is recovered in $O(n^\alpha \log n)$ observations. ■

It is worth noting that in this scenario, the attacker does not control the error. If the attacker controls the error locations, then it is possible to obtain x in $\lceil n/\varepsilon \rceil$ queries. This can happen during a fault attack, akin to side-channel attacks. It should also be noted that some coordinates of biometric data may be non-variable and, as a consequence, an attacker cannot recover them. This partial recovery attack is, therefore, a privacy attack, and leads to an authentication attack if the number of variable coordinates is sufficiently large (at least $n - \varepsilon$ in the binary case).

Remark 3.2.1. *In the non-binary case, the value $x_i - y_i$ does not provide enough information. The exact value of x_i can be determined in two cases. First, if $x_i - y_i = -q + 1$, then $x_i = 0$. Second, if $x_i - y_i = 2(q-1)$, then $x_i = q - 1$. For all other cases, there is an ambiguity regarding the value of x_i as y_i is unknown. However, by knowing the distribution of x_i and y_i, repeating observations yields a statistical attack.*

Attacks for each type of leakage along with their complexities are summarized in Figure 1.

4 CONCLUDING REMARKS

Our investigation into the information leakage of a biometric system using privacy-preserving distance has uncovered critical security vulnerabilities that arise under various scenarios. By evaluating the impact of different types of leakage, including distance, error position, and error value, we have highlighted the potential risks posed to data privacy and security.

Our analysis has encompassed 'below the threshold' and 'below and above the threshold' setups, allowing us to identify specific conditions under which information leakage can significantly affect the overall security of the system.

It is important to highlight that the leakage 'below the threshold' does not notably harm the security of the system, while the leakage of 'both below and above the threshold' markedly decreases the security. Indeed, the attacks exploiting the leakage 'below the threshold' are primarily exponential, while those exploiting information 'below and above the threshold' are mainly constant.

The accumulation attack we investigated assumes errors uniformly distributed throughout each authentication session. The result of the accumulation attack could be further refined by considering a variable number of coupons, randomly drawn between 0 and ε in each round, while acknowledging the actual distribution of the errors. To the best of our knowledge, no previous studies provide an analysis of the distribution of the errors for any systems.

In practical scenarios, certain errors may occur more frequently than others, while some may never occur. A skewed distribution of errors will substantially increase the expected number of authentications required from the legitimate user for the server to recover the hidden template in its entirety. Future research should involve refining the accumulation attack as suggested above and exploring other distance metrics, such as L_1 (*i.e.*, Manhattan distance) and L_2.

ACKNOWLEDGEMENTS

The authors acknowledge the support of the French Agence Nationale de la Recherche (ANR), under grant ANR-20-CE39-0005 (project PRIVABIO).

REFERENCES

Ahlgren, J. (2014). The probability distribution for draws until first success without replacement.

Aydin, F. and Aysu, A. (2024). Leaking secrets in homomorphic encryption with side-channel attacks. *Journal of Cryptographic Engineering*, pages 1–11.

Belguechi, R., Cherrier, E., Rosenberger, C., and Ait-Aoudia, S. (2013). Operational bio-hash to preserve privacy of fingerprint minutiae templates. *IET biometrics*, 2(2):76–84.

Berenbrink, P. and Sauerwald, T. (2009). The weighted coupon collector's problem and applications. In Ngo, H. Q., editor, *Computing and Combinatorics*, pages 449–458, Berlin, Heidelberg. Springer Berlin Heidelberg.

Bernal-Romero, J. C., Ramirez-Cortes, J. M., Rangel-Magdaleno, J. D. J., Gomez-Gil, P., Peregrina-Barreto, H., and Cruz-Vega, I. (2023). A review on protection and cancelable techniques in biometric systems. *IEEE Access*, 11:8531–8568.

Cho, S., Oh, B.-S., Kim, D., and Toh, K.-A. (2021). Palm-vein verification using images from the visible spectrum. *IEEE Access*, 9:86914–86927.

Chvatal, V. (1979). A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235.

Daugman, J. (2009). How iris recognition works. In *The essential guide to image processing*, pages 715–739. Elsevier.

Daugman, J. (2015). Information theory and the iriscode. *IEEE transactions on information forensics and security*, 11(2):400–409.

Dehkordi, A. B. and Abu-Bakar, S. A. (2015). Iris code matching using adaptive hamming distance. In *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 404–408. IEEE.

Ferrante, M. and Saltalamacchia, M. (2014). The coupon collector's problem. *MATerials MATemàtics*, 2014:35.

Harikrishnan, D., Sunil Kumar, N., Joseph, S., and Nair, K. K. (2024). Towards a fast and secure fingerprint authentication system based on a novel encoding scheme. *International Journal of Electrical Engineering & Education*, 61(1):100–112.

Hashemi, M., Forte, D., and Ganji, F. (2024). Time is money, friend! timing side-channel attack against garbled circuit constructions. In *International Conference on Applied Cryptography and Network Security*, pages 325–354. Springer.

He, R., Cai, Y., Tan, T., and Davis, L. (2015). Learning predictable binary codes for face indexing. *Pattern recognition*, 48(10):3160–3168.

Korte, B. H., Vygen, J., Korte, B., and Vygen, J. (2011). *Combinatorial optimization*, volume 1. Springer.

Ouda, O., Tsumura, N., and Nakaguchi, T. (2010). Bioencoding: A reliable tokenless cancelable biometrics scheme for protecting iriscodes. *IEICE TRANSACTIONS on Information and Systems*, 93(7):1878–1888.

Pagnin, E., Dimitrakakis, C., Abidin, A., and Mitrokotsa, A. (2014). On the leakage of information in biometric authentication. In *International Conference on Cryptology in India*, pages 265–280. Springer.

Patel, V. M., Ratha, N. K., and Chellappa, R. (2015). Cancelable biometrics: A review. *IEEE signal processing magazine*, 32(5):54–65.

Rahman, A., Chowdhury, M. E., Khandakar, A., Kiranyaz, S., Zaman, K. S., Reaz, M. B. I., Islam, M. T., Ezeddin, M., and Kadir, M. A. (2021). Multimodal eeg and keystroke dynamics based biometric system using machine learning algorithms. *Ieee Access*, 9:94625–94643.

Ratha, N. K., Connell, J. H., and Bolle, R. M. (2001). An analysis of minutiae matching strength. In Bigun, J. and Smeraldi, F., editors, *Audio- and Video-Based Biometric Person Authentication*, pages 223–228, Berlin, Heidelberg. Springer Berlin Heidelberg.

Sharma, S., Saini, A., and Chaudhury, S. (2023). A survey on biometric cryptosystems and their applications. *Computers & Security*, page 103458.

Simoens, K., Bringer, J., Chabanne, H., and Seys, S. (2012). A framework for analyzing template security and privacy in biometric authentication systems. *IEEE Transactions on Information Forensics and Security*, 7:833–841.

Thomas, M. and Joy, A. T. (2006). *Elements of information theory*. Wiley-Interscience.

Timothée, P. and Ramanna, S. C. (2016). *Tutorial 10 for Information Theory*.

Tran, L., Hoang, T., Nguyen, T., and Choi, D. (2017). Improving gait cryptosystem security using gray code quantization and linear discriminant analysis. In *International Conference on Information Security*, pages 214–229. Springer.

Wang, Z., Yang, J., and Zhu, Y. (2021). Review of ear biometrics. *Archives of Computational Methods in Engineering*, 28(1):149–180.

Yang, H. and Wang, Y. (2007). A lbp-based face recognition method with hamming distance constraint. In *Fourth international conference on image and graphics (ICIG 2007)*, pages 645–649. IEEE.