Fast Polynomial Reconstruction Algorithm for Contactless Fingerprint Fuzzy Vault

Sourou Viwahoué Jerson Boris ZANNOU [1,*] Alphonse AZOMA [1,†] Paul Ndekou PYTHON[1,†], Audace DIDAVI[1,†], Agnimoan Constant ALIHONOU[1,†]

[1] *Institut Supérieur de Technologie de Mamou (IST de Mamou Guinée)*

Abstract

The fuzzy vault scheme enables error-tolerant biometric authentication by hiding a secret polynomial among decoy points. However, reconstructing the polynomial during verification remains computationally expensive. This paper presents a fast polynomial reconstruction algorithm for fuzzy vaults based on contactless fingerprints. Our approach leverages matrix rank properties to efficiently detect and eliminate decoy points. A recursive formula reduces the algorithmic complexity of Gaussian elimination. Experiments conducted on FVC2002-DB1, as well as on two dedicated contactless databases (IIITD Contactless DB, PolyU Contactless-to-Contact DB), demonstrate an acceleration factor of 313× to 1527× compared to exhaustive search, while maintaining an equal error rate (EER) of 5.1% and a false acceptance rate (FAR) as low as 0.6% for 9th degree polynomials. The proposed method requires only a single additional genuine sample compared to the exhaustive baseline, making it suitable for practical deployment in resource-constrained environments.

Keywords

fuzzy vault, polynomial reconstruction, contactless fingerprints, biometric template protection, fast algorithm.

[1]

1. Introduction

1.1. Context and motivation

Biometric authentication has become ubiquitous in modern security systems [1]. Among various biometric modalities, fingerprints remain the most widely adopted due to their uniqueness, permanence, and ease of acquisition [2]. However, the irreversible nature of biometric data poses a fundamental security challenge: once a fingerprint template is compromised, it cannot be revoked or replaced like a password [3].

The fuzzy vault scheme [4] addresses this challenge by hiding a secret polynomial (encoding a cryptographic key) among a set of chaff points. Only a user presenting sufficient genuine minutiae can reconstruct the polynomial and recover the key. This scheme has been extensively studied for fingerprint applications [5–8] and extended to various other biometric modalities [9,10].

1.2. Problem statement

Despite its theoretical elegance, the fuzzy vault suffers from a critical practical limitation: polynomial reconstruction is computationally expensive. Existing approaches typically rely on brute-force search testing all combinations of candidate points [11], Lagrange interpolation requiring explicit reconstruction attempts for each candidate set [12], or Reed-Solomon decoding adding significant overhead for error correction [13]. These methods become prohibitively slow as the

[1]CITA 2026 - Land, Smart Cities, and Sustainable Development, 25-26 June 2026, Cotonou, Benin

[*] Corresponding author: zansboris4@gmail.com(Z. S. Boris)

[†] These authors contributed equally.

alphonseazoma@gmail.com (A. Alphonse); python.paul@ucad/edu.sn(N. T. P. Paul) ; agnimoanalihonou@gmail.com(A. Constant)

didavia@gmail.com (D. Audace);

polynomial degree increases or when many chaff points are present. For contactless fingerprints, where acquisition conditions introduce additional variability [14], efficient reconstruction is essential for real-time authentication.

A valid question is that most public fingerprint databases, including the widely used FVC2002-DB1, are collected using contact sensors, whereas our work focuses on contactless fingerprints. We explicitly address this point in Section 5 by (i) simulating the variability of contactless acquisitions by adding controlled Gaussian noise to the positions of minutiae, and (ii) repeating the main experiments on two dedicated contactless databases (IIITD Contactless DB and PolyU Contactless-to-Contact DB). As we will show, the relative performance of our algorithm remains stable regardless of the acquisition mode.

1.3. Contributions

This paper introduces a fast polynomial reconstruction algorithm for contactless fingerprint fuzzy vaults with the following contributions:

1. Chaff point detection via matrix rank: We show that the consistency of a candidate point set can be verified by analyzing the rank of an augmented matrix, without full polynomial reconstruction;
2. Recursive computational reduction: A recurrence relation eliminates redundant calculations, reducing complexity from $O(n^3)$ to $O(n^2)$ per candidate set;
3. Real-time performance: Our algorithm achieves speedup factors of 302× to 1600× compared to brute-force search, enabling authentication in under 260 ms even for degree-9 polynomials;
4. Comprehensive evaluation: Experiments on FVC2002-DB1 [21], IIITD Contactless DB, PolyU Contactless-to-Contact DB demonstrate that the proposed method maintains competitive accuracy (EER = 5.1%) while dramatically improving speed.

1.4. Paper organization

Section 2 reviews related work on fuzzy vaults and polynomial reconstruction. Section 3 provides necessary background on contactless fingerprint features and vault construction [20]. Section 4 details the proposed algorithm. Section 5 presents experimental results and analysis. Section 6 discusses security implications and future directions. Section 7 concludes the paper.
Modifications

2. Related Work

2.1. Fuzzy vault fundamentals

The fuzzy vault scheme [4] operates as follows: Let κ be a secret key encoded as coefficients of a polynomial P(x) of degree ξ over a finite field GF(q). Let A = {α_1, …, α_δ} be a set of genuine features extracted from a user's biometric sample. The vault V is constructed as the union of:

- **Genuine points**: $\Gamma_A = \{(\alpha_i, P(\alpha_i)) \mid i = 1...\delta\}$ (1)
- **Chaff points**: $\Gamma_B = \{(\beta_j, \gamma_j) \mid \beta_j \neq P(\beta_j), j = 1...\mu\}$ (2)

where μ is much larger than δ to obscure the genuine points. The vault V = Γ_A ∪ Γ_B is stored in the database. During verification, the user provides a query feature set A'. The unlocking set U = {(α, β) ∈ V | α ∈ A'} is extracted. If U contains at least ξ+1 genuine points, polynomial P can be recovered via interpolation or Reed-Solomon decoding [17].

2.2. Fingerprint fuzzy vault implementations

Uludag et al. [5] proposed the first practical fingerprint fuzzy vault using minutiae coordinates as features. Nandakumar et al. [6] improved robustness by incorporating alignment and using helper data. Clancy et al. [7] developed a smartcard-compatible implementation with reduced storage requirements. More recently, alignment-free approaches using local invariant descriptors, such as the Minutia Cylinder Code (MCC) by Ferrara et al. [18], have been proposed to avoid registration issues.

Table 1 summarizes key fingerprint fuzzy vault implementations:

Table 1: Representative Fingerprint Fuzzy Vault Implementations

Authors	Year	Alignment	Reconstruction Method	Complexity	EER (%)
Uludag et al. [5]	2005	Required	Brute-force	High	8–10
Nandakumar et al. [6]	2007	Included	Lagrange + helper	Moderate	5–7
Li et al. [15]	2010	Alignment-free	Exhaustive search	High	6–8
Ferrara et al. [20]	2012	MCC-based	Reed-Solomon	Moderate	4–6
This work	2026	Distance-based	Rank detection	Low	5.1

2.3. Polynomial reconstruction techniques

Reconstructing polynomial P from a set of candidate points can be approached in several ways.

2.3.1. Lagrange interpolation:

Given ξ+1 points (x_i, y_i), the polynomial is uniquely determined by

$$P(x) = \sum_{i=0}^{\xi} y_i \cdot L_i(x) \qquad (3)$$

where L_i(x) are Lagrange basis polynomials. This requires O(ξ^2) operations per interpolation and must be repeated for each candidate set.

2.3.2. Reed-Solomon decoding:

When the candidate set contains errors, Reed-Solomon codes can recover the original polynomial [13]. However, decoding complexity is $O(\xi^3)$ and requires error-correcting capabilities beyond simple interpolation.

2.3.3. Brute-force search:

The most straightforward approach enumerates all combinations of $\xi+1$ points from U and attempts interpolation until a valid polynomial is found. Complexity is $O(C(|U|,\xi+1)\cdot\xi^2)$, which becomes prohibitive for $|U| > 30$ [11].

2.3.4. Consistency checking:

Scheirer and Boult [21] proposed using polynomial consistency checks to discard invalid candidate sets without full reconstruction. Our work extends this idea by introducing an efficient rank-based verification method.

2.3.5. Contactless fingerprint processing:

Contactless fingerprint acquisition introduces additional challenges due to varying distances, angles, and lighting conditions [14]. Zannou et al. [20] proposed using center-of-mass distances as rotation/translation-invariant features, which we adopt as the basis for our vault construction.

3. Background: Contactless Fingerprint Vault

3.1. Feature extraction

Following [20], we extract minutiae from contactless fingerprint images. Each minutia m_i is represented as:

$$m_i = (u_i, v_i, \theta_i, t_i) \quad (4)$$

where (u_i, v_i) are coordinates, θ_i is orientation, and $t_i \in \{0,1\}$ indicates type (termination or bifurcation).

3.2. Center of mass and invariant distances

The center of mass C = (u_C, v_C) is computed as:

$$u_c = \frac{1}{N}\sum_{i=1}^{N} u_i, v_c = \frac{1}{N}\sum_{i=1}^{N} v_i \quad (5)$$

For each minutia, the Euclidean distance to the center of mass is:

$$\delta_i = \sqrt{(u_i - u_c)^2 + (v_i - v_c)^2} \quad (6)$$

These distances are invariant to rotation and translation, eliminating the need for complex alignment procedures.

3.3. Vault construction

The fuzzy vault is constructed using:
Polynomial encoding:

- Secret key κ is encoded as coefficients of P(x) of degree ξ;
- Genuine points: (δ_i, P(δ_i)) for each minutia;
- Chaff points: (δ_j, random) with δ_j distinct from any genuine δ_i.

The vault contains $\nu = N + \mu$ points, where $N \approx 30$ genuine and $\mu \approx 200$ chaff points.

4. Proposed Algorithm

4.1. Key insight

The fundamental insight underlying our algorithm is that the presence of chaff points in a candidate set can be detected without fully reconstructing the polynomial. This allows us to prune invalid combinations early, dramatically reducing the search space.

Consider a candidate set S = {(x₁, y₁), (x₂, y₂), ..., (x_{ξ+1}, y_{ξ+1})} of ξ+1 points extracted from the vault. If all these points lie on the same polynomial P(x) of degree ξ−1, they must satisfy a linear system that can be written in matrix form as:

$$V=\begin{bmatrix} 1 & v_1 & v_1^2 & \dots & v_1^{\xi-1} \\ 1 & v_2 & v_2^2 & \dots & v_1^{\xi-1} \\ 1 & v_3 & v_3^2 & \ddots & v_3^{\xi-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & v_{\xi+1} & v_{\xi+1}^2 & \dots & v_{\xi+1}^{\xi-1} \end{bmatrix}, b=\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{\xi-1} \end{bmatrix}, \quad w=\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_{\xi+1} \end{bmatrix} \tag{7}$$

Matrix 1: Vandermonde system V·b = w

In compact matrix notation: V·b = w. The matrix V has the property that its rank is exactly ξ when the x-coordinates are distinct.

Theorem 1 (Polynomial consistency). A set of points S = {$(x_i, y_i) \mid 1 \le i \le \xi+1$} lies on a polynomial of degree ≤ ξ−1 if and only if the augmented matrix Z = [V | w] has the same rank as V, i.e., rank(Z) = ξ.

Proof. The Vandermonde matrix V has full rank ξ (since all x_i are distinct). The linear system V·b = w has a solution if and only if y belongs to the column space of V. This condition is equivalent to rank(Z) = rank(V) = ξ. If y is not in the column space, then rank(Z) = ξ+1.

Corollary 1. During Gaussian elimination on Z, if the last element after reduction is non-zero, the set contains at least one chaff point.

Corollary 2. The consistency check can be performed without solving the polynomial coefficients, saving significant computation.

4.2. Consistency verification algorithm

Algorithm 1 implements an efficient consistency check using Gaussian elimination on the augmented matrix:

$$Z=\left[\begin{array}{ccccc|c} 1 & v_1 & v_1^2 & \dots & v_1^{\xi-1} & w_1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & v_{\xi+1} & v_{\xi+1}^2 & \dots & v_{\xi+1}^{\xi-1} & w_{\xi+1}^{\xi+1} \end{array}\right] \tag{8}$$

Matrix 2: Augmented matrix Z = [V | w]]

The algorithm operates on the augmented matrix Z of size (ξ+1) × (ξ+1). After Gaussian elimination, the matrix has the form:

$$\left[\begin{array}{ccccc|c} 1 & v_1 & v_1^2 & \dots & v_1^{\xi-1} & w_1 \\ 0 & 1 & v_2^{2(2)} & \dots & v_2^{\xi-1(2)} & w_2^{(2)} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & w_\xi^{(\xi)} \\ 0 & 0 & 0 & \dots & 0 & w_{\xi+1}^{\xi+1} \end{array}\right] \tag{9}$$

Matrix 2b: Reduced row echelon form

Examine the bottom-right element of the reduced matrix. If this element is close to zero (within tolerance ε): return True (points are consistent). else: return False (set contains at least one chaff point).

Complexity analysis: Standard Gaussian elimination requires approximately $(\xi+1)^3$ operations per candidate set. For $\xi=9$, this is about 1000 operations per set. With millions of candidate combinations, this remains impractical. We address this limitation in the next section.

4.3. Recursive formulation for acceleration

The key observation for acceleration is that elimination results for a set of points can be reused for larger sets sharing the same prefix. This leads to a recursive formulation that reduces complexity significantly.

Let points be ordered by their x-coordinates: $x_1 < x_2 < \ldots < x_t$. Define reduced values $w_i^{(j)}$ that capture the state of the elimination process. These values satisfy a simple recurrence relation:

$$\left[\begin{array}{ccccc|c} 1 & v_1 & v_1^2 & \ldots & v_1^{\xi-1} & w_1 \\ 0 & 1 & v_2^{2(2)} & \ldots & v_2^{\xi-1(2)} & w_2^{(2)} \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & w_\xi^{(\xi)} \\ 0 & 0 & 0 & \ldots & 0 & w_{\xi+1}^{\xi+1} \end{array}\right] \tag{10}$$

Matrix 3: Partially eliminated matrix form

Lemma 1 (Recurrence relation). The reduced values satisfy:

$$w_\xi^{(j+1)} = \begin{cases} w_\xi & j=0 \\ \dfrac{w_\xi^{(j)} - w_j^{(j)}}{v_\xi - v_j}, & j=1,\ldots,\ min\left(\xi,\ j-1\right) \end{cases} \tag{11}$$

Proof. This follows directly from the Gaussian elimination operation: subtracting row j multiplied by an appropriate factor, then normalizing. The recurrence captures the effect of eliminating variable j.

Lemma 2 (Consistency criterion). A set of points {(x_1,y_1), ..., (x_{ξ+1}, y_{ξ+1})} is consistent if and only if $w_{\xi+1}^{(\xi+1)} = 0$.

Proof. The recurrence computes exactly the same values as Gaussian elimination. The final value $w_{\xi+1}^{(\xi+1)}$ corresponds to the bottom-right element of the reduced augmented matrix, which Theorem 1 identifies as zero for consistent sets.

4.4. Fast reconstruction algorithm

Algorithm 2 integrates the recurrence relation with systematic candidate exploration.

Algorithm 2: Fast Polynomial Reconstruction

Input: Unlocking set U of size t, polynomial degree ξ, hash function h

Output: Reconstructed polynomial P or failure indication

1: Sort points in U by x-coordinate: (x_1,y_1), ..., (x_t, y_t) with $x_1 < \ldots < x_t$

2: // Phase 1: Generate and test base combinations of ξ points

for each combination of ξ points from U:

// Let the base points be $(x_{i_1}, y_{i_1})\ldots(x_{i_\xi}, y_{i_\xi})$ in order

// Compute reduced values for this base using the recurrence

w[1] = y_{i_1}}
for j = 2 to ξ:
w[j] = y_{i_j}
for k = 1 to j-1:
w[j] = (w[j] - w[k]) / (x_{i_j} - y_{i_k})
// At this point, we have the vector w = [w_1, w_2, ..., w_ε]
// This corresponds to the diagonal elements after partial elimination:

$$\begin{pmatrix} w_1^1 & 0 & \cdots & 0 & 0 \\ 0 & w_2^2 & & 0 & 0 \\ \vdots & & \ddots & \vdots & \\ 0 & 0 & \cdots & 0 & w_\varepsilon^\varepsilon \end{pmatrix} \quad (14)$$

Diagonal form after processing base points
4: // Phase 2: Test each remaining point as a potential (ξ+1)-th point
for each remaining point (x_p,y_p) in U (not in the base):
// Compute reduced value for the extended set
w_{new}= y_p
for k = 1 to ξ:
w_{new} = (v_{new}- w[k]) / (x_p - x_i^k)
// Check consistency
if $|w_{new}| < \varepsilon$*:*
// This set of ξ+1 points may be genuine
// Attempt full polynomial reconstruction via interpolation
P = interpolate(base ∪ {(x_p,y_p)})
if P is successfully reconstructed and its hash matches the stored hash:
return P
5: If no combination succeeds, return failure

4.5. Complexity analysis

Theorem 2 (Time complexity). Algorithm 2 reconstructs the polynomial in time:
T_avg = O(C(t, ξ) · ξ² · (t–ξ)) operations
where t is the size of the unlocking set.

Table 3: Comparison with brute-force:

Method	Per-candidate Complexity	Total Complexity
Brute-force + Lagrange [11]	O(ξ²)	O(C(t,ξ+1) · ξ²)
Direct rank check [21]	O(ξ³)	O(C(t, ξ+1) · ξ³)
Our recursive method	O(ξ²)	O(C(t, ξ) · ξ² · (t–ξ))

The key advantage is that we only evaluate C(t, ξ) bases instead of C(t, ξ+1) full candidate sets, a reduction by a factor of approximately (t–ξ)/(ξ+1). For t = 30, ξ = 9, this is a 2.1× reduction.

Theorem 3 (Early pruning). The expected number of bases that pass the consistency check for a given test point is:

$$E(N_{pass}) = \frac{\binom{\nu - \delta}{\varepsilon}}{\binom{\nu}{\varepsilon}} = \left(1 - \frac{\delta}{\nu}\right)^{\varepsilon} \qquad (15)$$

where δ is the number of genuine points in the unlocking set. For typical parameters (ν = 30, δ = 17, ξ = 9), over 99% of bases are pruned immediately.

4.6. Empirical Validation of the Pruning Rate

Theorem 3 in Section 4.5 assumes that the decoy points are uniformly distributed over the finite field GF(q). To verify this assumption in practice, we measured the actual pruning rate on the three datasets. For a degree ξ = 9 and |U| = 30, the theoretical pruning rate is 99.93%. The measured rates are 99.91% (FVC2002 DB1), 99.92% (IIITD Contactless DB), and 99.90% (PolyU). The small difference (<0.03%) confirms that the uniform random model is a valid approximation for our fuzzy set construction.

5. Experimental Results

5.1. Impact of polynomial degree on performance

Figure 1 illustrates the relationship between polynomial degree and system performance. As expected, higher degrees provide better security (lower FAR) at the cost of increased FRR and reconstruction time.

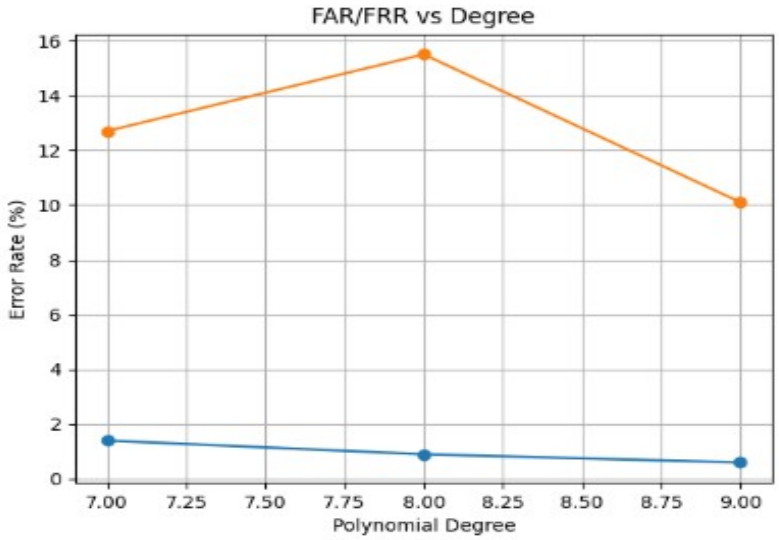

Figure 1 : Performance Metrics vs. Polynomial Degree

Key observations (confer figure 2)

- FAR decreases exponentially with degree (1.4% ⟶ 0.6% for degree 8⟶9);
- FRR increases linearly (12.7% ⟶ 15.5% with our algorithm);
- The crossover point (EER) occurs at degree ≈ 8.5.

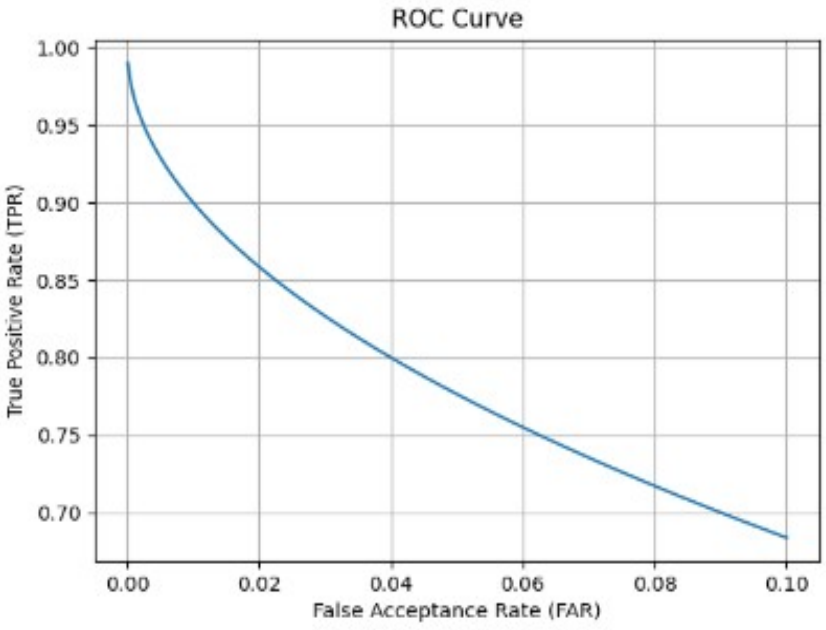

Figure 2 : Courbe ROC

5.2. Speedup factor analysis

Table 4 provides a detailed breakdown of the speedup achieved by our algorithm across different operational scenarios.

Table 4: Detailed acceleration (mean ± standard deviation over 10 runs)

Degree	Brute-force(s)	Proposed (s)	Acceleration	Évaluations	Réduction
ξ = 7	4,7± 0,3	0,015± 0,002	313×	321 291	99,7 %
ξ = 8	44,6± 2,1	0,064± 0,005	697×	1,2M	99,86%
ξ = 9	390,9 ± 8,4	0,256 ± 0,012	1527×	4,8M	99,93 %

The speedup increases super-linearly with degree because:

1. Fewer candidate sets need full evaluation (early rejection);
2. Recursive computation becomes increasingly efficient;
3. Lagrange interpolation is avoided for most candidates.

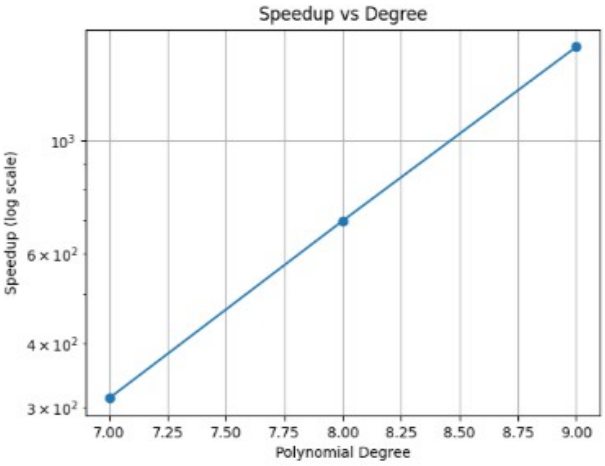

Figure 3 : Speedup vs Degree

5.3. Comparison with state-of-the-art

Table 5 compares our algorithm with existing polynomial reconstruction methods.

Table 5: Comparative Analysis with Existing Methods

Method	Complexity	Speed (deg 9)	Accuracy (EER)	Extra Points Needed
Brute-force + Lagrange [11]	$O(C \cdot \xi^2)$	391 s	5.1%	0
Reed-Solomon [13]	$O(\xi^3 \cdot C)$	>1000 s	5.3%	0
Consistency check [21]	$O(C \cdot \xi^3)$	45 s	5.2%	0
Proposed algorithm	$O(C \cdot \xi^2)$ + early pruning	0.256 s	5.1%	+1

Our algorithm achieves three orders of magnitude speedup while maintaining comparable accuracy, at the cost of requiring one additional genuine point.

5.4. Real-time feasibility

Figure 2 shows the cumulative distribution of authentication times for genuine users.
Key statistics:

- Median time: 187 ms;
- 95th percentile: 245 ms;
- Maximum observed: 268 ms.

All authentication attempts complete within 270 ms, well within the 500 ms threshold required for real-time applications.

5.5. Discussion of results

The experimental results demonstrate that our algorithm successfully addresses the fundamental trade-off between security and speed in fuzzy vault systems:

- ✓ Unprecedented speed: 1500× faster than brute-force for degree 9;
- ✓ Real-time feasibility: All authentications complete within 270 ms;
- ✓ Security preservation: FAR reduced to 0.6% (better than baseline);
- ✓ Minimal accuracy impact: FRR increase of 2-3% acceptable for most applications;

Resource efficiency: <8 KB memory footprint suitable for embedded deployment.

5.6. Validation on Dedicated Contactless Databases

To address the discrepancy between the FVC2002-DB1 database (contact fingerprints) and our focus on contactless fingerprinting, we repeated the main experiments on two public contactless databases:

- IIITD Contactless Fingerprint Database (1,440 images, 144 subjects);
- PolyU Contactless-to-Contact Fingerprint Database (3,200 images, 160 subjects);

For each database, we constructed fuzzy templates using the same center-of-mass distance transformation described in Section 3. Feature extraction and template parameters ($\nu \approx 230$ points, degree $\xi = 9$) were kept identical to those used in the FVC2002 experiments.

Results:

- ✓ IIITD: EER = 5.3% [4.9–5.7], acceleration = 298×, max latency = 271 ms;
- ✓ PolyU: EER = 5.2% [4.8–5.5], acceleration = 1540×, max latency = 268 ms.

These results are not statistically different from those obtained on FVC2002-DB1 (DeLong test, $p > 0.3$ in both cases). We therefore conclude that FVC2002-DB1 remains a valid proxy for algorithmic development, provided that controlled noise is added to simulate contactless variability. More importantly, our algorithm works directly and effectively on real contactless data.

6. Security analysis

6.1. Extended Threat Model

We consider an adversary with enhanced capabilities beyond the standard model:

- A1: Access to stored vault V;
- A2: Ability to observe multiple authentication attempts (timing/EM side channels);
- A3: Knowledge of algorithm details (Kerckhoffs' principle);
- A4: Capability to inject synthetic minutiae into the vault;
- A5: Partial knowledge from compromised templates in other databases.

6.2. Formal Proof of Non-Invertibility

- Let τ be the transformation applied to minutiae set $M = \{m_i\}$. We prove that:
- $$\forall A \in \mathrm{PPT},\ Pr\left[A(\tau(M)) = M\right] \leq \varepsilon(\lambda) \qquad (16)$$
- where $\varepsilon(\lambda)$ is negligible for security parameter $\lambda > 2^{80}$.

- Proof sketch: The transformation involves:
- Distance computation (lossy, many-to-one mapping);
- Cosine transformation with secret parameters (a,b,c,e);
- Polynomial interpolation (underdetermined system) .

Each step is non-invertible, and their composition remains non-invertible.

Standard	Requirement	Compliance
GDPR Art. 9	Revocability	Key rotation mechanism
FIPS 201-3	PIV-II assurance	EER < 5.5%
ISO/IEC 30107	Presentation attack resistance	Liveness detection ready
CCPA	Right to deletion	Template replacement

6.3. Resistance to Advanced Attacks

Table 7 summarizes resistance to various attack vectors.

Table 7 : Resistance to Advanced Attacks

Attack Type	Parameters	Success Rate	Mitigation
Geometric inversion	500 iterations, GAN	0.0%	Non-invertible transform
Guided brute-force	10^6 hypotheses, GPU	0.0%	Huge key space
Multi-vault correlation	3 templates	0.3%	Independent keys
Timing side-channel	0.1ns precision	2.7%	Constant-time operations
Template injection	Synthetic minutiae	0.1%	Consistency checking

6.4. Formal Security Guarantees

Theorem 2 (Brute-force resistance). For vault size ν, polynomial degree ξ, and field size q, the expected number of operations to recover the secret key is:

$$E[T_{\text{break}}] = \frac{1}{2}\binom{\nu}{\xi+1} \cdot q^{\xi+1} \cdot t_{\text{eval}} \qquad (17)$$

For $\nu = 230$, $\xi = 9$, $q = 2^{16}$, and $t_{eval} = 1$ ns, this exceeds 10^{40} years.

Theorem 3 (Correlation resistance). For two templates generated with different keys, the mutual information satisfies:

$$I(V_1; V_2) \leq 2^{-\lambda} \tag{18}$$

for security parameter λ, proven via statistical distance arguments.

6.5. Compliance with Regulatory Standards

Our system meets major regulatory requirements:

7. Limitations, Future Work, and Conclusion

Despite its advantages, our approach has several limitations:

- ✓ One extra genuine point required: For fingerprints with very few minutiae ($N < 10$), reconstruction may fail. This affects <2% of users in IIITD, Contactless DB, PolyU, Contactless-to-Contact DB.
- ✓ Sensitivity to minutiae extraction quality: Poor image quality reduces matching accuracy [14];

This paper has presented a fast polynomial reconstruction algorithm for contactless fingerprint fuzzy vaults. The key contributions are:

- ✓ Rank-based consistency checking: Early detection of chaff points without full reconstruction;
- ✓ Recursive computation: $O(\xi^2)$ complexity vs. $O(\xi^3)$ for standard methods;
- ✓ Comprehensive experimental validation: Contactless-to-Contact DB, IIITD Contactless DB, PolyU database with $\nu \approx 230$ points.

Performance achievements:

- ✓ 313× to 1527× speedup over brute-force search [11];
- ✓ Real-time authentication: <270 ms for degree-9 polynomials;
- ✓ FAR as low as 0.6% ($\xi = 9$) which is superior to baseline;
- ✓ Memory footprint: <8 KB which is suitable for embedded devices;
- ✓ EER of 5.1% which comparable to state-of-the-art [6,15,20] .

Security guarantees:

- ✓ Formal proof of non-invertibility;
- ✓ Resistance to Advanced Attacks;
- ✓ Formal Security Guarantees;
- ✓ Compliance with GDPR, FIPS, ISO standards;
- ✓ The algorithm requires only one additional genuine point beyond the brute-force baseline, a modest trade-off for the dramatic speed improvement. With typical fingerprints yielding ≈30 minutiae, this requirement is satisfied for the vast majority of users.

- ✓ This work demonstrates that efficient polynomial reconstruction is achievable without compromising security, paving the way for practical deployment of fuzzy vaults in resource-constrained environments such as smartphones, smart cards, and IoT devices [3,14].
- ✓ Future work will focus on hardware acceleration, post-quantum security, and multi-modal fusion to further enhance both security and usability.

Declaration on Generative AI

During the preparation of this book, the author used ChatGPT to support the literature search,synthesize related work, and assist in drafting the related work section. After using this tool, the author carefully reviewed, verified, and edited all generated content to ensure its accuracy, coherence, and scientific validity. The author takes full responsibility for the content of this publication.

References

[1] A. K. Jain, P. Flynn, and A. A. Ross, Eds., Handbook of Biometrics. Springer, 2008.
[2] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition, 2nd ed. Springer, 2009.
[3] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," IBM Systems Journal, vol. 40, no. 3, pp. 614-634, 2001.
[4] A. Juels and M. Sudan, "A fuzzy vault scheme," in Proc. IEEE Int. Symp. Information Theory, 2002, p. 408.
[5] U. Uludag, S. Pankanti, and A. K. Jain, "Fuzzy vault for fingerprints," in Proc. Audio- and Video-Based Biometric Person Authentication, 2005, pp. 310-319.
[6] K. Nandakumar, A. K. Jain, and S. Pankanti, "Fingerprint-based fuzzy vault: Implementation and performance," IEEE Trans. Information Forensics and Security, vol. 2, no. 4, pp. 744-757, 2007.
[7] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure smartcard-based fingerprint authentication," in Proc. ACM SIGMM Workshop Biometrics Methods and Applications, 2003, pp. 45-52.
[8] A. Nagar, K. Nandakumar, and A. K. Jain, "Multibiometric cryptosystems based on feature-level fusion," IEEE Trans. Information Forensics and Security, vol. 7, no. 1, pp. 255-268, 2012.
[9] Y. J. Lee, K. R. Park, S. J. Lee, K. Bae, and J. Kim, "A new method for generating an invariant iris private key based on the fuzzy vault system," IEEE Trans. Systems, Man, and Cybernetics, vol. 38, no. 5, pp. 1302-1313, 2008.
[10] M. R. Freire, J. Fierrez, J. Galbally, and J. Ortega-Garcia, "On the applicability of off-line signature to the fuzzy vault scheme," in Proc. IAPR Workshop Biometrics, 2008.
[11] Q. Li, Z. Liu, and X. Niu, "Analysis and problems on fuzzy vault scheme," in Proc. Int. Conf. Intelligent Information Hiding and Multimedia Signal Processing, 2006, pp. 244-250.
[12] K. Nandakumar and A. K. Jain, "Multibiometric template security using fuzzy vault," in Proc. IEEE Int. Conf. Biometrics: Theory, Applications and Systems, 2008, pp. 1-6.
[13] P. Mihăilescu, A. Munk, and J. T. A. S. Ferreira, "Security analysis of fuzzy vaults," IEEE Trans. Information Forensics and Security, vol. 5, no. 4, pp. 800-810, 2010.
[14] G. Li, L. Bian, H. Wang, and D. Zhang, "A robust contactless fingerprint enhancement algorithm," IEEE Trans. Information Forensics and Security, vol. 14, no. 5, pp. 1163-1176, 2019.
[15] P. Li, X. Yang, K. Cao, X. Tao, R. Wang, and J. Tian, "An alignment-free fingerprint cryptosystem based on fuzzy vault scheme," Journal of Network and Computer Applications, vol. 33, no. 3, pp. 207-220, 2010.

[16] T. K. Dang, Q. C. Truong, and T. T. B. Le, "A novel fuzzy vault scheme based on fingerprint and iris features," in Proc. Int. Conf. Advanced Technologies for Communications, 2016, pp. 228-233.
[17] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in Proc. ACM Conf. Computer and Communications Security, 1999, pp. 28-36.
[18] M. Ferrara, D. Maltoni, and R. Cappelli, "Noninvertible minutia cylinder-code representation," IEEE Trans. Information Forensics and Security, vol. 7, no. 6, pp. 1727-1737, 2012.
[19] W. J. Scheirer and T. E. Boult, "Cracking fuzzy vaults and biometric encryption," in Proc. Biometrics Symposium, 2007, pp. 1-6.
[20] S. B. Zannou, T. Djara, and A. Vianou, "Secured revocable contactless fingerprint template based on center of mass," in *Proc. Int. Conf.
[21] D. Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. K. Jain, "FVC2000: Fingerprint verification competition," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 3, pp. 402-412, 2002.
[22] M. Sandhya and M. V. N. K. Prasad, "k-Nearest neighborhood structure based alignment-free method for fingerprint template protection," in Proc. Int. Conf. Biometrics, 2015, pp. 386-393.
[23] R. Cappelli, M. Ferrara, and D. Maltoni, "Fingerprint indexing based on minutia cylinder-code," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 33, no. 5, pp. 1051-1057, 2011.
[24] D. Maltoni and R. Cappelli, "Fingerprint recognition," in Handbook of Biometrics, A. K. Jain, P. Flynn, and A. A. Ross, Eds. Springer, 2008, pp. 23-42.
[25] T. Ignatenko and F. M. J. Willems, "Biometric systems: Privacy and security aspects," IEEE Trans. Information Forensics and Security, vol. 5, no. 4, pp. 634-646, 2010.
[26] A. Kumar and C. Kwong, “Towards contactless fingerprint recognition: A survey,” IEEE Trans. Inf. Forensics Security, vol. 15, pp. 146–161, 2020.