Towards Fingerprints as Strings: Secure Indexing for Fingerprint Matching

Dr. Venu Govindaraju  
(joint work with Atri Rudra)  
University at Buffalo
Biometric Data

- Store and Retrieve data

**String**

login, password → 6A03...F322

**Cryptographic Hash**

002D...8CB9
0053...0427
...
6A03...F322
...
FF68...381A

**Lexicographic Database**

- **Strings:** Privacy preserving AND Indexed hashes
- **Fingerprints:** ??
Competing Objectives

- Indexing and fast retrieval
- Privacy preserved (non-invertible)

Traditional Databases:
- Fast retrieval
- Not secure

Privacy Preserving Databases:
- Slow retrieval
- Secure
Problem Statement

Design a secure, fast, and accurate methodology for large biometric databases

Privacy Preserving methods

P1: Error correcting codes
Fuzzy commitment, fuzzy vault, secure sketches

P2: Many-to-one hash functions:
Random projections, symmetric polynomial hashes
P1: Error Correcting Code Methods

- Fuzzy vault - Juels, Sudan [2002];
- Applications for fingerprints - Clancy et al.[2003]

\[ p(x) = k_n x^n + \ldots + k_0 \]

Secret k

Locking set of points

Chaff points

Database

Unpacking set of points

Reconstruct polynomial
Assume Reed-Solomon codes. Welch-Berlekamp algorithm will succeed in decoding $P_k$ (order $k$) and reconstructing the polynomial, if retrieved set of pairs $\{(x,y)\}$ has $t$ (of $n$) pairs satisfying $(y=P(x))$, and $t>(n+k)/2$ ($t>\sqrt{nk}$) Guruswamy and Sudan 1999.

$$p(x) = k_n x^n + \ldots + k_0$$

Secret $k$

Locking set of points

Database

Unlocking set of points

Reconstruct polynomial

Chaff points
P2: Projection Methods

Fingerprint enrollment

Image → Minutiae → Hashes

Fingerprint verification

Image → Minutiae → Hashes

Store → Fingerprint Hash Database

Match
**Enrolled hashes of minutia n-plet:**

\[ h_1 = c_1 + c_2 + \ldots + c_n \]

\[ h_2 = c_1^2 + c_2^2 + \ldots + c_n^2 \]

\[ \vdots \]

\[ h_m = c_1^m + c_2^m + \ldots + c_n^m \]

***Hashes of rotated/translated n-plet (test print); Not invertible if m<n:***

\[ h'_1 = (rc_1 + t) + (rc_2 + t) + \ldots + (rc_n + t) = rh_1(c_1, c_2, \ldots, c_n) + nt \]

\[ h'_2 = (rc_1 + t)^2 + \ldots + (rc_n + t)^2 = r^2h_2(c_1, c_2, \ldots, c_n) + 2rth_1(c_1, c_2, \ldots, c_n) + nt^2 \]
P2: Projection Methods

- Can calculate transformation parameters \( r, t \) given \( h_1, h_2, h'_1, h'_2 \)
- Find \( r, t \) for all possible \( n \)-plets of enrolled and test fingerprint
- Global match = find the cluster of similar transformation parameters
## Problem Statement

Design a secure, fast, and accurate methodology for large biometric database

### Fast Indexing methods

<table>
<thead>
<tr>
<th></th>
<th>Method</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>Pyramid indexing</td>
<td>Mehtre et al 2004</td>
</tr>
<tr>
<td>I2</td>
<td>Feature set indexing</td>
<td>Germain 1997; Tan 2003</td>
</tr>
<tr>
<td>I3</td>
<td>Feature path indexing</td>
<td>Mansukhani et al 2010</td>
</tr>
</tbody>
</table>
I2: Feature Set Indexing Method

Bag of minutia triplets; each triplet is associated with some bin
Store and match by columns only as table is sparse
Count fingerprints with most matches: 1 (2 times), 3 (1 time), 5 (1 time)

I3: Feature Path Indexing Methods
(IEEE Systems Journal 2010)

• Paths in neighboring minutia = branches in indexing tree

<table>
<thead>
<tr>
<th>Templates Searched</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.56</td>
<td>0.74</td>
<td>0.80</td>
<td>0.86</td>
<td>0.87</td>
</tr>
<tr>
<td>Matching Rate</td>
<td>0.32</td>
<td>0.63</td>
<td>0.76</td>
<td>0.84</td>
<td>0.85</td>
</tr>
</tbody>
</table>
Competing Objectives

- Use extra data (eg, orientation)
- Retain explicit structure (minutiae triplets) information
- Don’t leak private information
- Obfuscate (chaff points)
- Discard data (projection)
- What data must be retained to keep balance?
- How can you be fast?
Theoretical Privacy Measure (Fuzzy Vault)

\[ \text{Cost} = C_1 FRR + C_2 FAR + C_3 \lambda + C_4 T \]

\[ \lambda = \sqrt{rk - t} \]

- \( r \) – total # of points (genuine and chaff)
- \( k-1 \) – degree of encoding polynomial
- \( t \) - # of genuine points in a vault

\[ \lambda = \sqrt{rk - t} \]

Privacy measure

Matching/enrollment time

- Assume cryptographic hardness of Reed-Solomon codes (Kiayias and Yung, 2008)
- Assume list decoding algorithm (Guruswami and Sudan, 1999)
Result 1
(SPIE 2013)

\[ Cost = C_1 FRR + C_2 FAR + C_3 \lambda \]

- Experimental evaluation (fingerprint FVC2002 database)
- Average no of true minutiae positions \( \sim 45 \)
Result 2: Fuzzy Vault + Indexing (P1+I2)  
(ICB 2013)

- Use paths of 5 nearest minutia; each path corresponds to a bin
- For each enrolled template $i$ extracted bin $j$ add polynomial $P_i(b_j)$
- Add random chaff points with random polynomial values

Matching:

<table>
<thead>
<tr>
<th>Enrolled templates</th>
<th>1, $P_1(b_1)$</th>
<th>2, $P_2(b_2)$</th>
<th>2, $P_2(b_4)$</th>
<th>1, $P_c$</th>
<th>1, $P_1(b_9)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2, $P_2(b_2)$</td>
<td>2, $P_2(b_4)$</td>
<td></td>
<td>2, $P_2(b_8)$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3, $P_3(b_2)$</td>
<td></td>
<td>3, $P_3(b_5)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>4, $P_4(b_3)$</td>
<td>4, $P_4(b_6)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5, $P_c$</td>
<td>5, $P_5(b_4)$</td>
<td></td>
<td>5, $P_5(b_9)$</td>
<td></td>
</tr>
</tbody>
</table>

- Retrieved indexes: 1, 5, 3, 1, 4, 5
- Count fingerprints with most matches: 1 (2 times), 3 (1 time), 4 (1 time), 5 (2 times)
- Perform fuzzy vault error correction for each matched fingerprint
Result 2: Fuzzy Vault + Set Indexing (P1+I2 + I3)

Indexing performance / Penetration rate

Time to search the indexing table: (x-axis is exponential scale)
Result 2:  Fuzzy Vault + Set Indexing (P1+I2+I3)

Matching and security performance on FVC2002

<table>
<thead>
<tr>
<th></th>
<th>DB1</th>
<th>DB2</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>( \lambda )</td>
<td>FAR</td>
</tr>
<tr>
<td>2</td>
<td>72.0</td>
<td>0.2386</td>
</tr>
<tr>
<td>3</td>
<td>143.2</td>
<td>0.3654</td>
</tr>
<tr>
<td>4</td>
<td>203.3</td>
<td>0.4816</td>
</tr>
</tbody>
</table>
Summary

• Improved security hurts the indexing efficiency and accuracy

• Alignment avoided by using translation/rotation free features which probably led to lower matching accuracy

• Indexing efficiency affected by the additional chaff points

• Need to derive privacy and indexing measures for other techniques

• Which algorithms can be combined for minimum loss of security and efficiency?
References