ICB 2007

# Robust Extraction of Secret Bits from Minutiae

Ee-Chien Chang

School of Computing
National University of Singapore

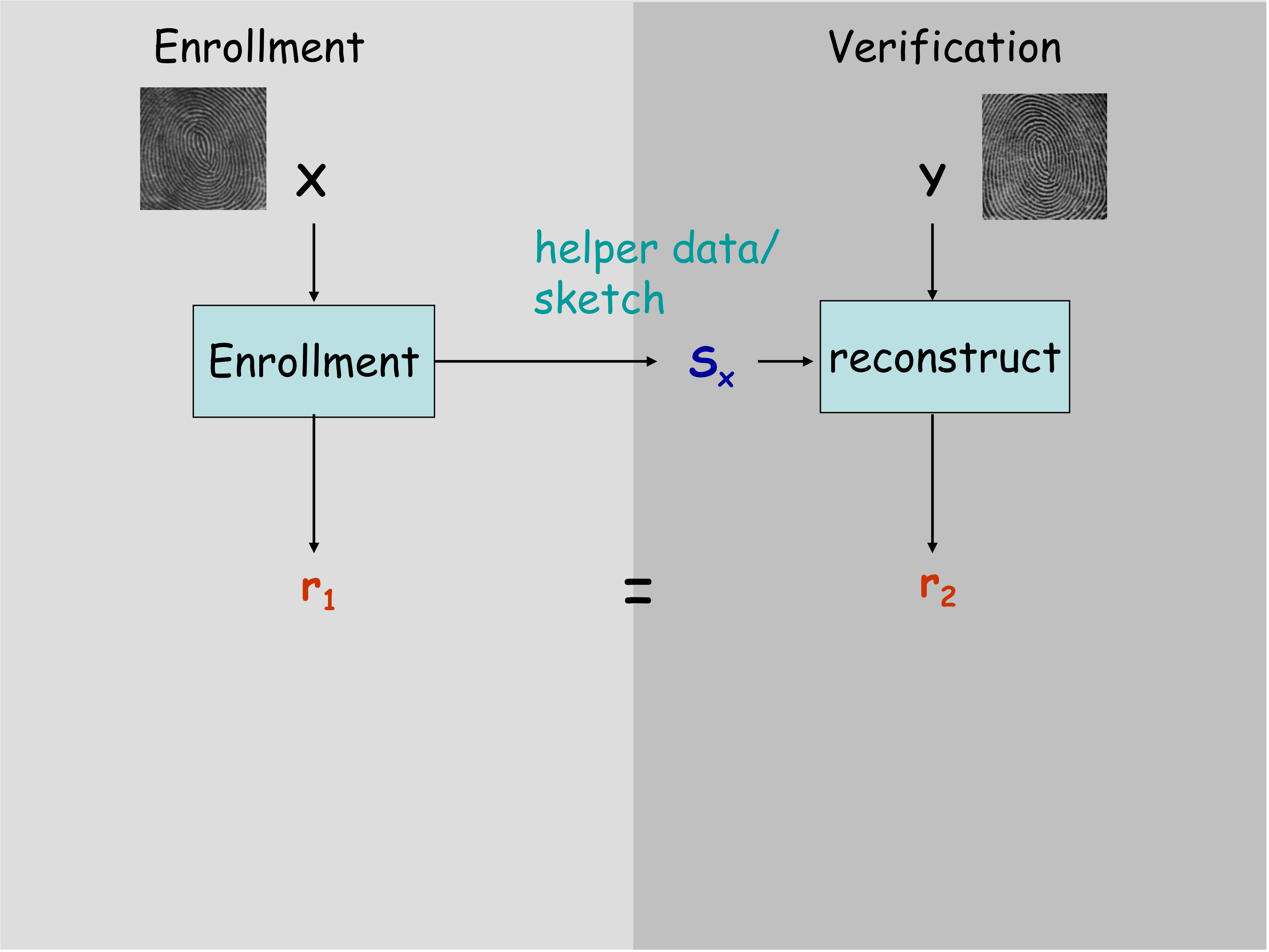
Sujoy Roy

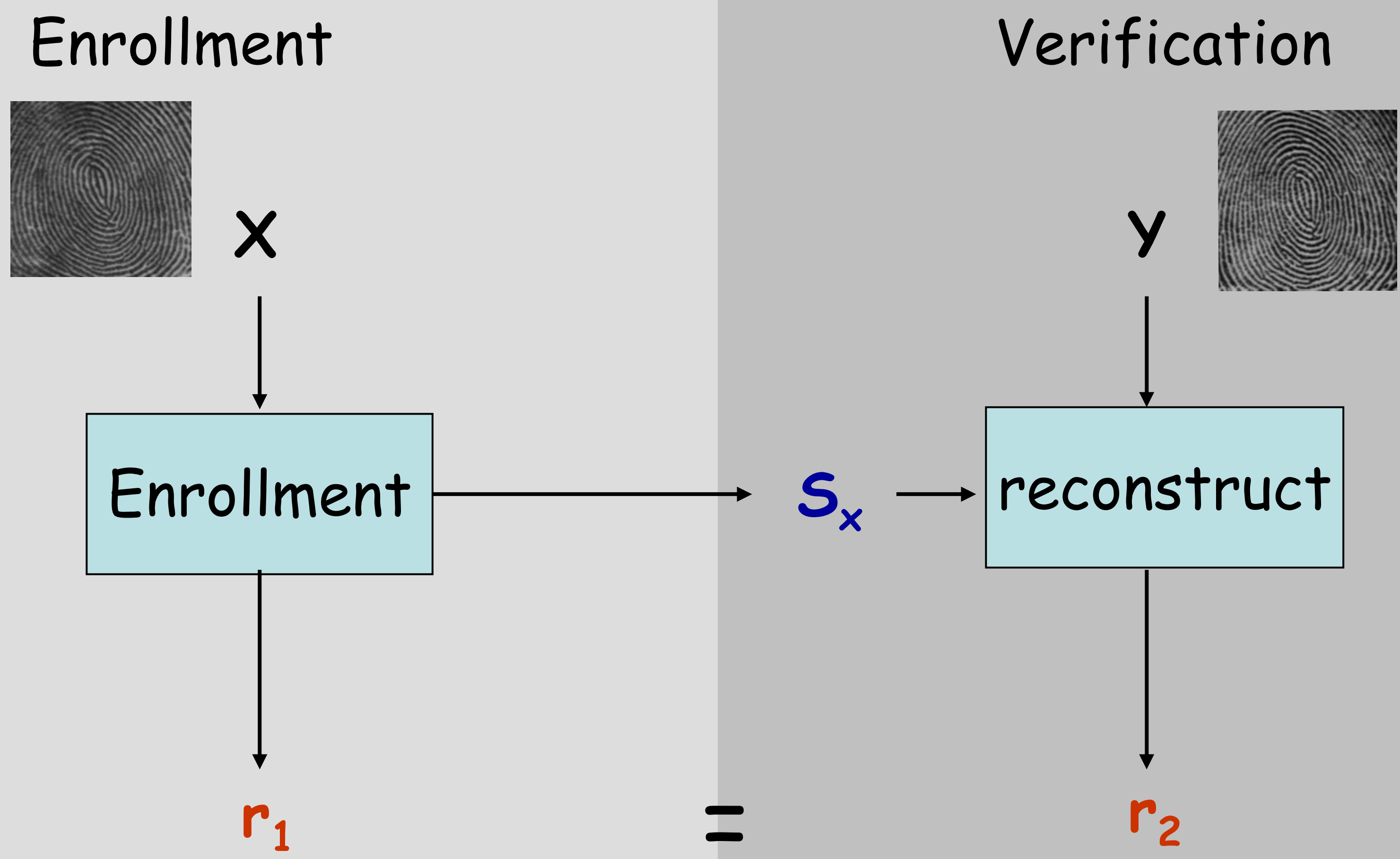
Institute for Infocomm Research
Singapore

# Motivation

- To extract consistent bits from different scans of a same finger. From two different scans, the extracted bits must be ***exactly*** the same.

- Such bits can be used as the secret in cryptographic applications.

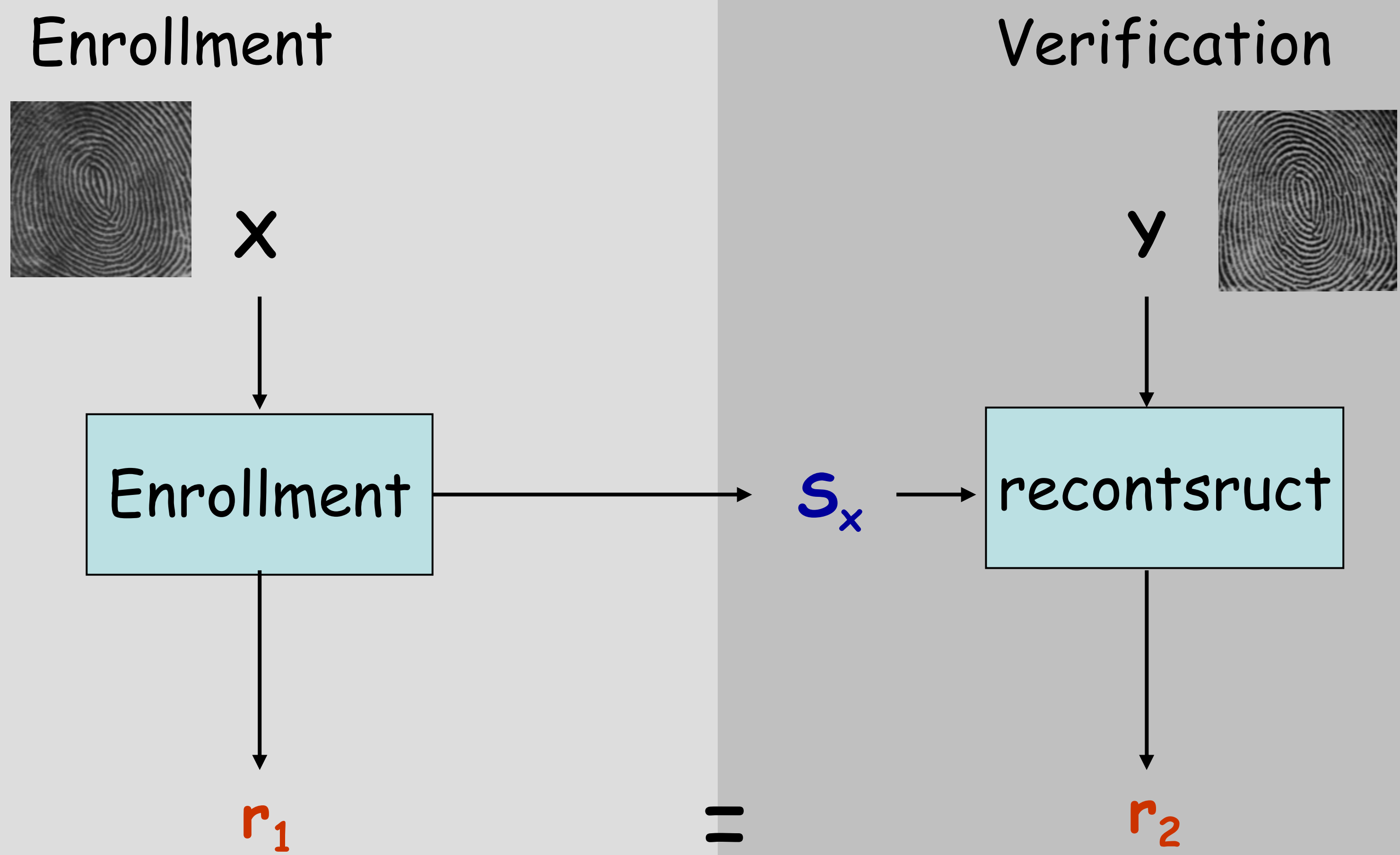

Enrollment
Verification
x
y
helper data/
sketch
Enrollment
$s_x$
reconstruct
$r_1$
=
$r_2$




- $S_x$ may reveal some information of $r_1$ and $S_x$ must be made public.

- entropy of secret bits.

  $H( r_1 \mid S_x )$

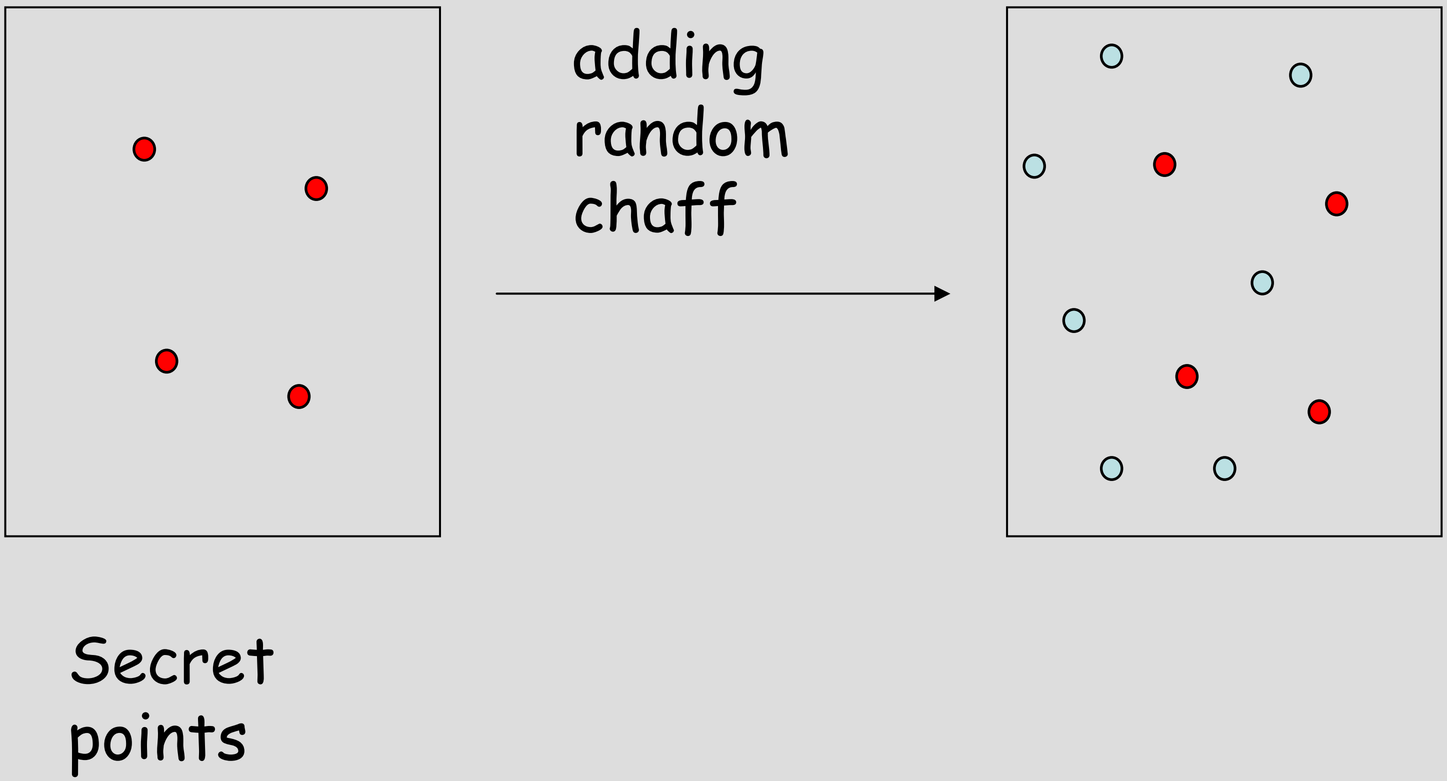



- The above framework can employed for biometric authentication.

  Thus $H(r_1 \mid S_x) \leq -\log_2 \text{FMR}$

So, if the state-of-the-art authentication system achieves FMR=0.001. Probably we can't extract more than 10 bits.

# Chaff-based method



# Chaff-based method


Secret points

Form one part of the sketch.
This is made public

## Limitations of chaff-based methods

- Large sketch size.
- Inflexible to incorporate statistical properties of the data and noise.
- Difficult to give a statement on its security.

## Our approach

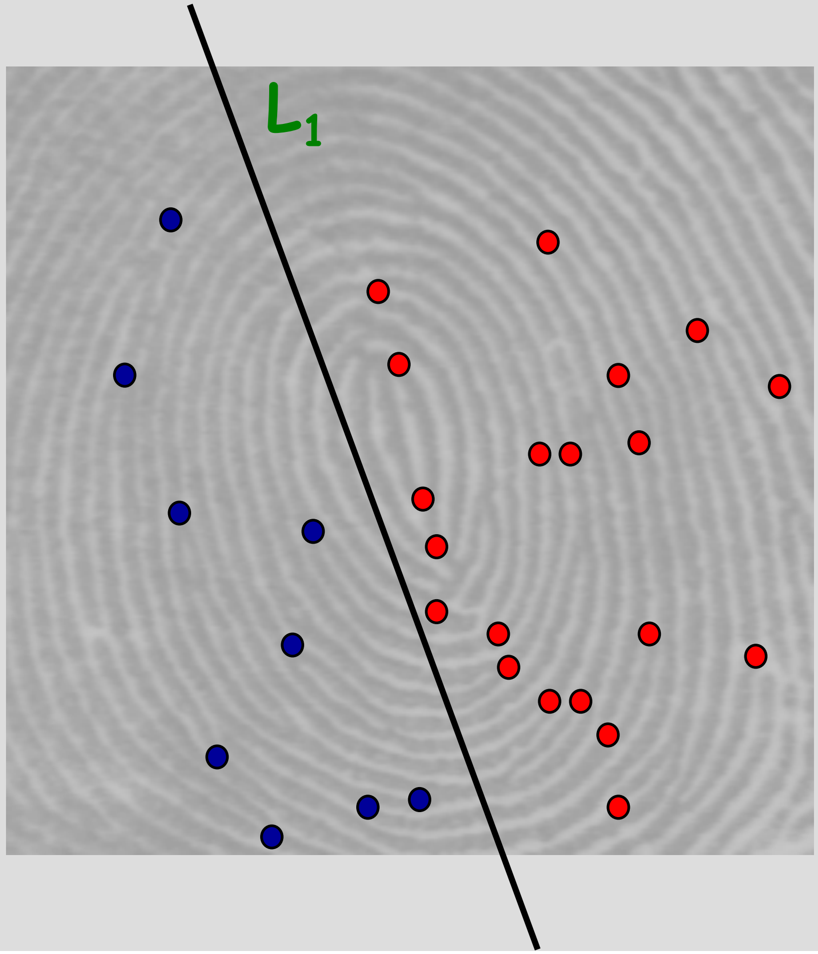
- Employ a locality preserving function to map the set of minutiae to a real vector.
- Using error-correcting codes on binary string to construct the sketch.

# Mapping minutiae to real vector

Choose many lines (for e.g. 600).

For a given line, and a set of minutiae X,
determine the different of the number of minutiae
on the left and right

9-20 = -11



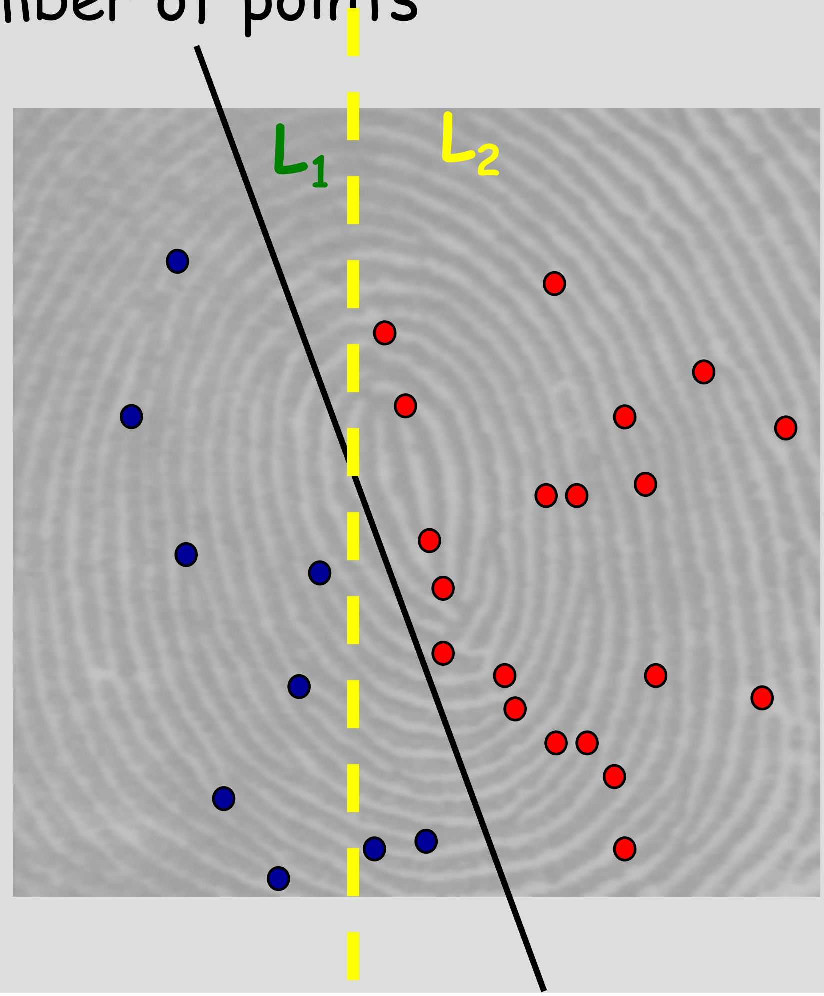
# Mapping minutiae to real vector

Choose many lines (for e.g. 600).

For a given line, and a set of minutiae X, determine the different of the number of points on the left and right

9-20 = -11

7-22 = -15



# Enrollment

- Map the minutiae $X$ to a real vector.

$$X \rightarrow v_x$$

- De-correlate and keep $k$ coefficients (PCA during design stage).

$$v_x \rightarrow h_x$$

- Convert to a $k$-bits string $b_x$.

$$h_x \rightarrow b_x$$

- Find the nearest codeword in a codebook of size $2^m$.

$c$ = nearest codeword to $b_x$.

- Compute sketch

$$s_x = b_x \text{ .xor. } c$$

- The secret bits are c or the message associated with c

# Enrollment

- Map the minutiae $X$ to a real vector.

  $X \rightarrow v_x$

- De-correlate and keep $k$ coefficients (PCA during design stage).

  $v_x \rightarrow h_x$

- Convert to a $k$-bits string $b_x$.

  $h_x \rightarrow b_x$

Assume that the $b_x$ are uniformly distributed.

- Find the nearest codeword in a codebook of size $2^m$.

  $c$ = nearest codeword to $b_x$.

- Compute sketch

  $s_x = b_x$ .xor. $c$

- The secret bits are c or the message associated with c

# Verification

- Same as enrollment, obtain a k-bits string $b_y$.
- compensate for noise using sketch

$$c = b_y \text{ .xor. } s_x$$

- Maximum likelihood decoding to find the "enrolled" codeword. (nearest codeword w.r.t. to a weighted Hamming distant derived from statistical properties of noise).

# Experiment

- We use NIST 4 database (2000 fingers with 2 scans each).

  100 pairs for training.

  - PCA
  - the weights in the weighted Hamming distance

- Using random codebook.

  (for different parameters, **k**, **m**)

# Performance

**Parameters:**

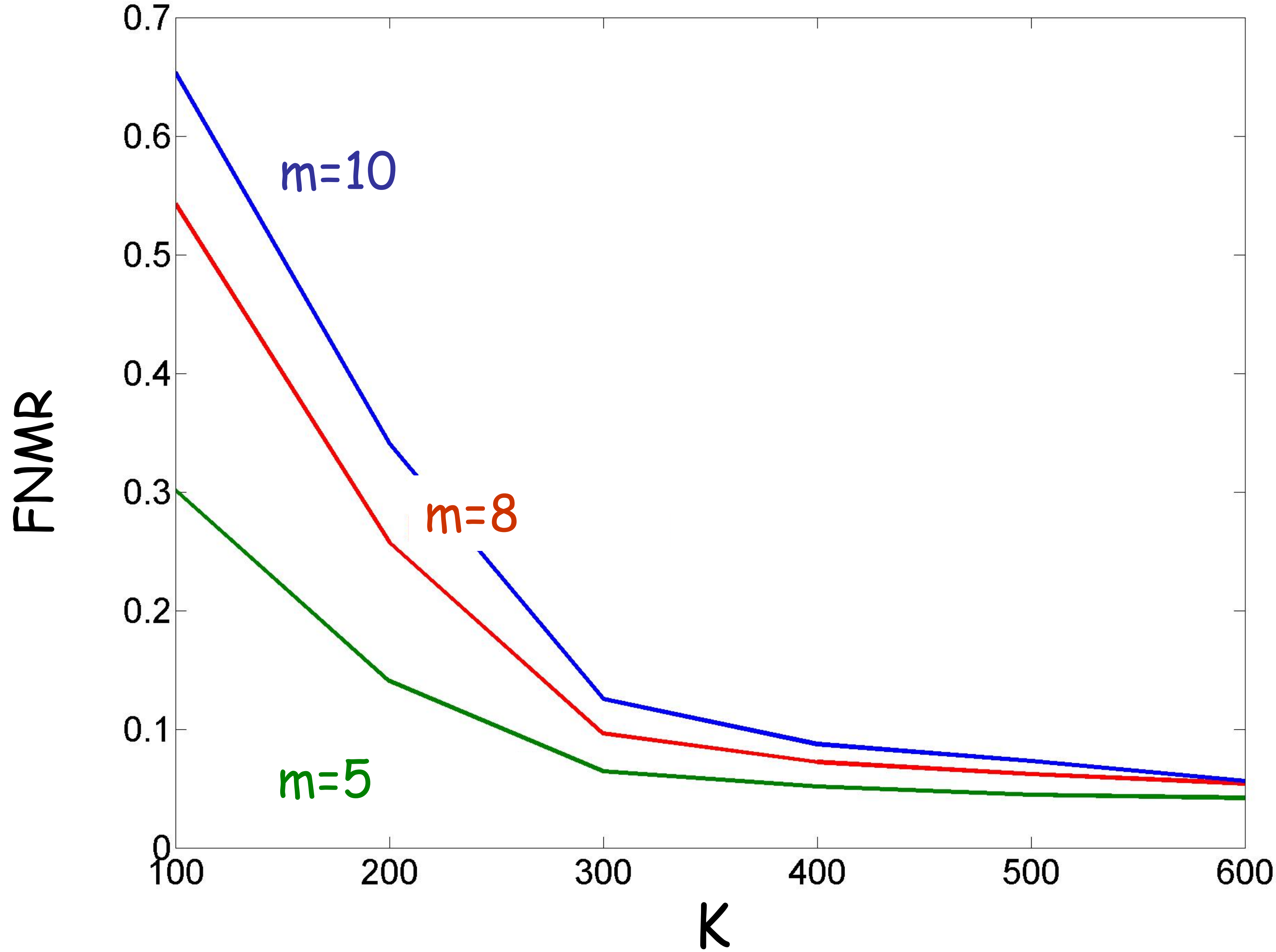
1. k: Number of coefficients retained after PCA.

2. m: $2^m$ is the number of codewords in the codebook.
   (number of secret bits, $-\log_2$ FMR)

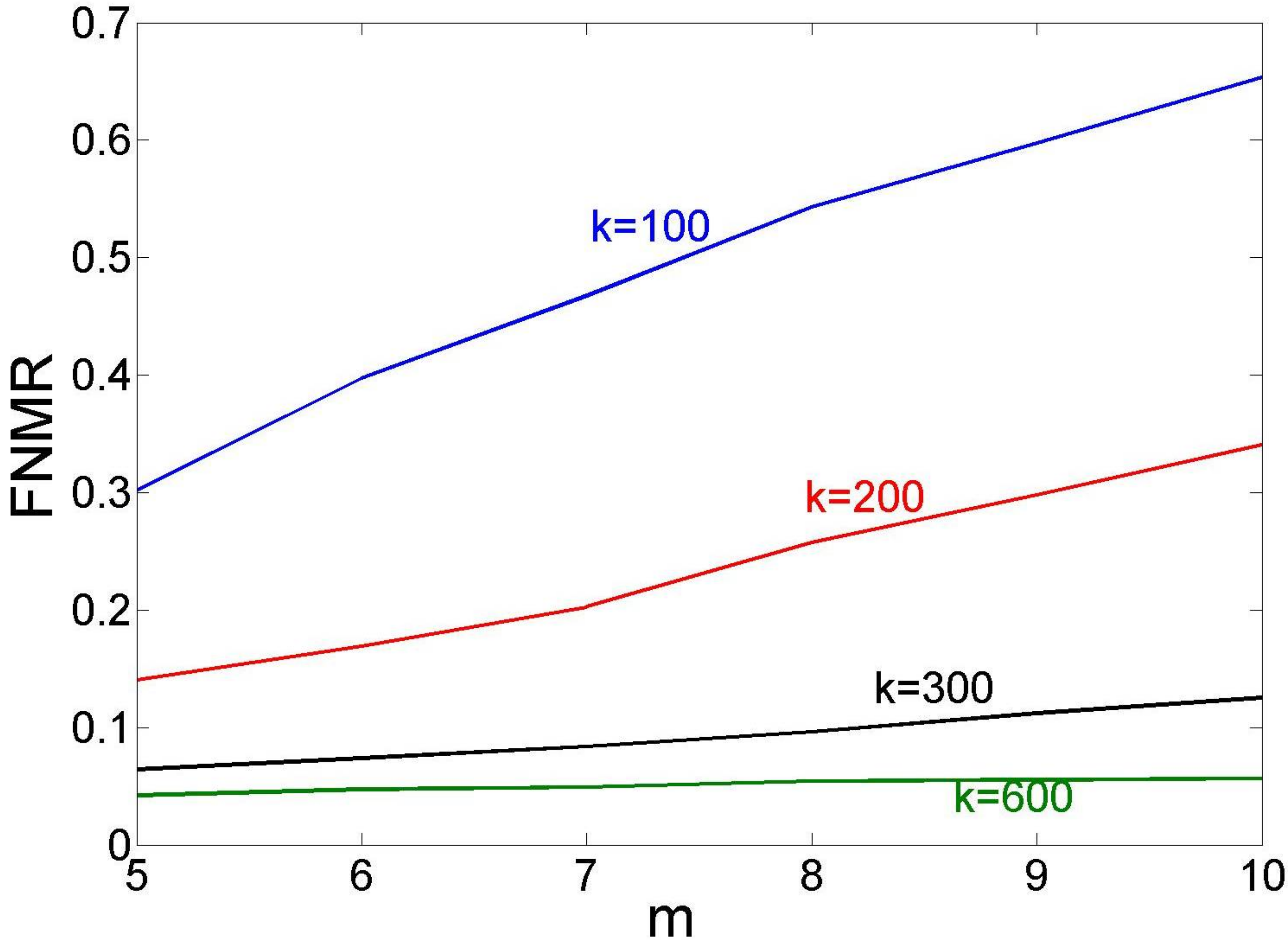
**Performance measures:**

1. FNMR

2. Size of the sketch.



m=10
m=8
m=5
FNMR
K
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
100
200
300
400
500
600



FNMR
m
k=100
k=200
k=300
k=600
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
5
6
7
8
9
10

# Conclusion

- Short sketch. (≈320 bits, no randomness)
- Able to incorporate statistical properties of minutiae. (PCA)
- Able to incorporate statistical properties of noise. (Maximum likelihood decoding)
- Able to make a statement on the number of secret bits.
  - At most 320 bits revealed.
  - If an intermediate representation is uniform distributed, then the number of secret bits is ≈10.

# Corrections

Change occurrences of

"FNMR 0.09%" → "FNMR 0.09"