

Distributed Source Coding for Secure Biometrics

Anthony Vetro

In collaboration with:

Current Team

Stark C. Draper (U. Wisconsin)
Jonathan S. Yedidia (MERL)
Yagiz Sutcu (Polytechnic U.)
Shantanu Rane (MERL)

Past Contributors

Emin Martinian (Bain Capital)
Ashish Khisti (MIT)

DISCOVER WORKSHOP
“Recent Advances in Distributed Video Coding”
November 6, 2007 Lisbon, Portugal

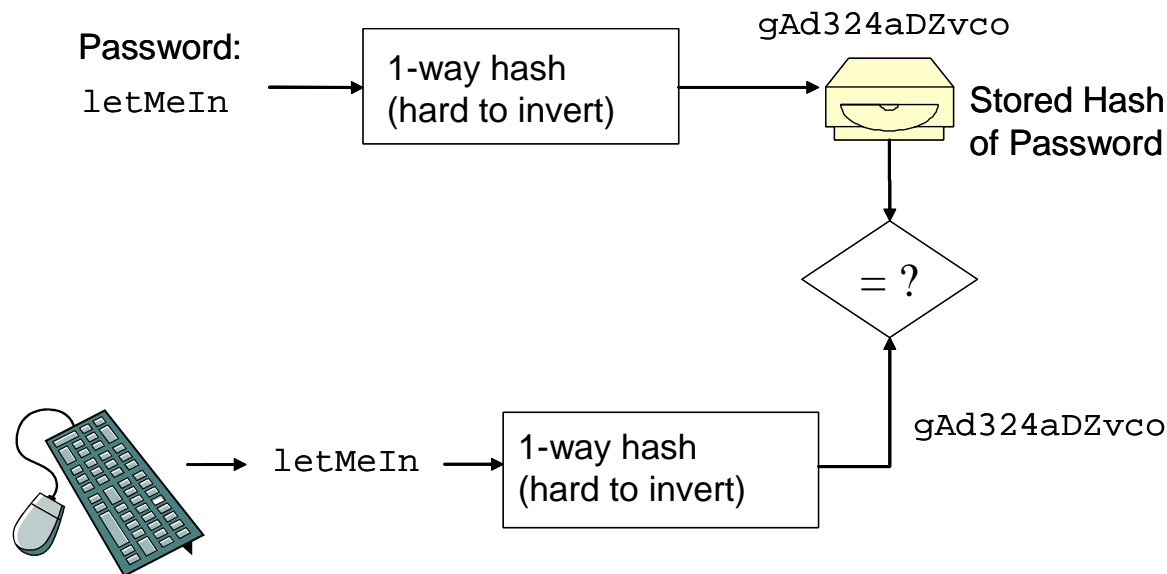
Motivation: Current biometric storage insecure

Password-based access

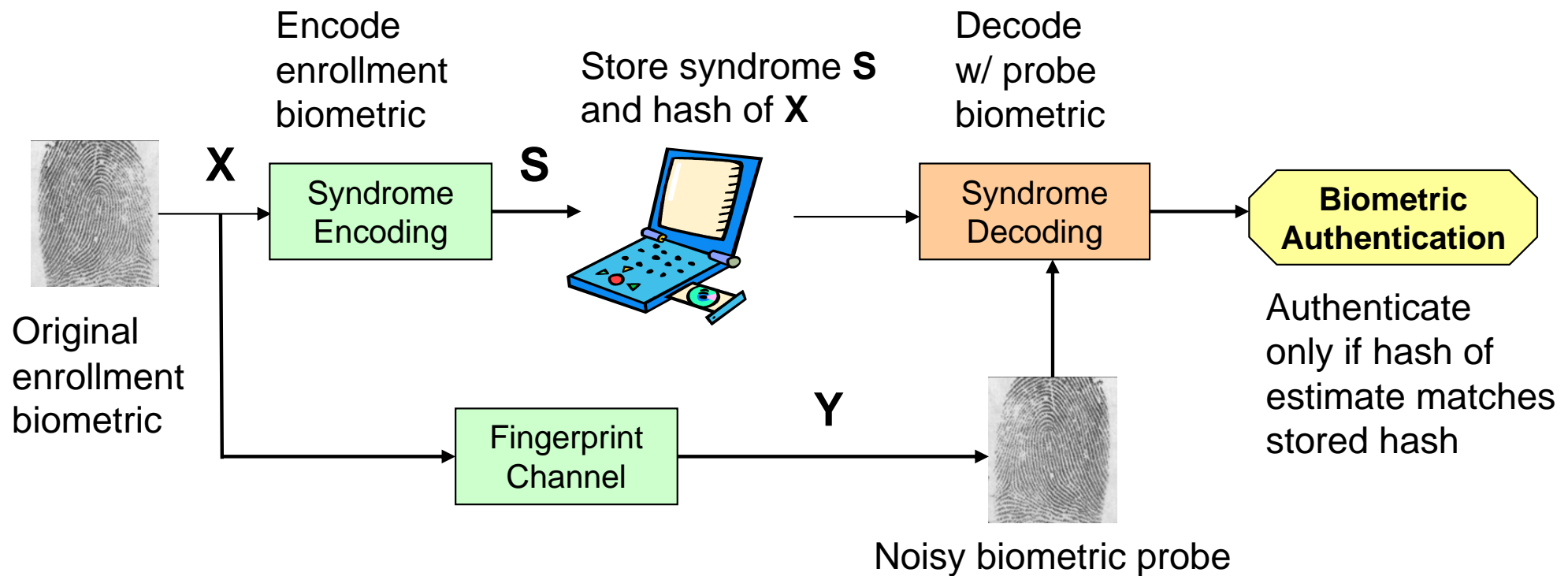
- Do **not** store passwords as clear text
- Store hash of password
- **If computer stolen / broken into, password remains secure**
- Enter identical password to gain access

Biometric-based access

- Store biometric as clear text and pattern match
- **Stolen computer = lost biometric**
- Biometric readings vary between measurements
- Results of a hash not repeatable



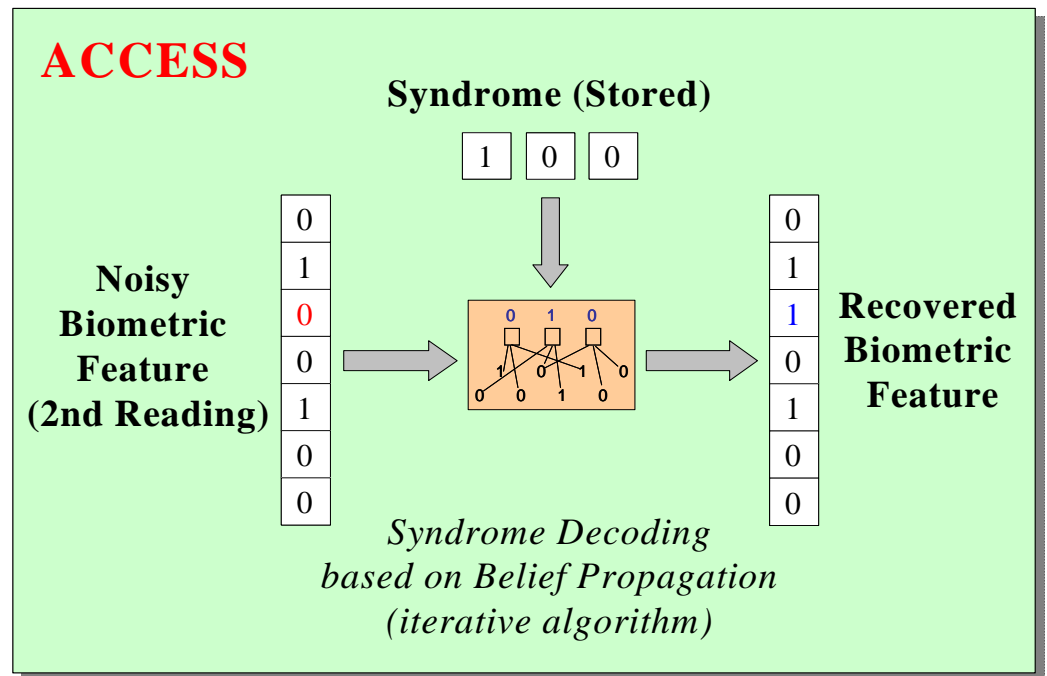
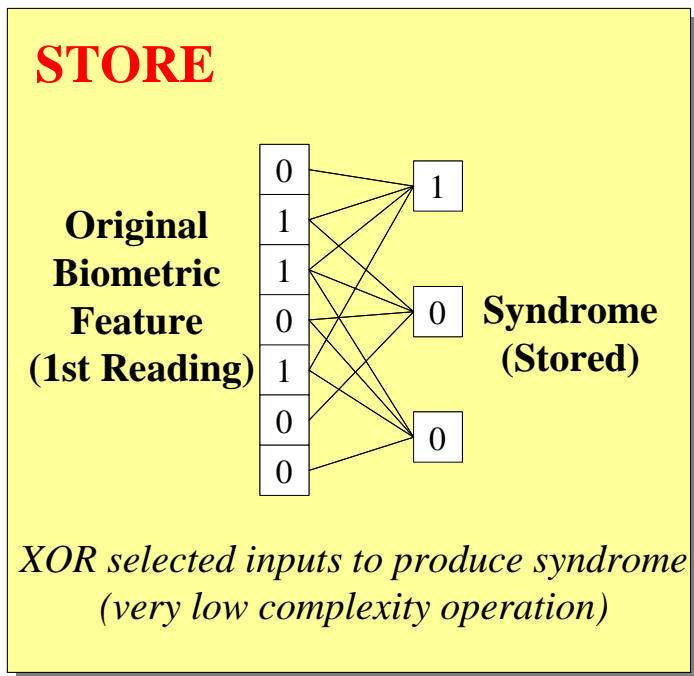
Conceptually just a Slepian-Wolf system



Encode into syndrome **S**

- **S** cannot be uncompressed by itself & is therefore secure
- In combination with a noisy second reading **Y** the original **X** can be recovered using a Slepian-Wolf decoder
- Compare hash of estimate with stored hash to permit access

Overview: Syndrome encoding / SW decoding

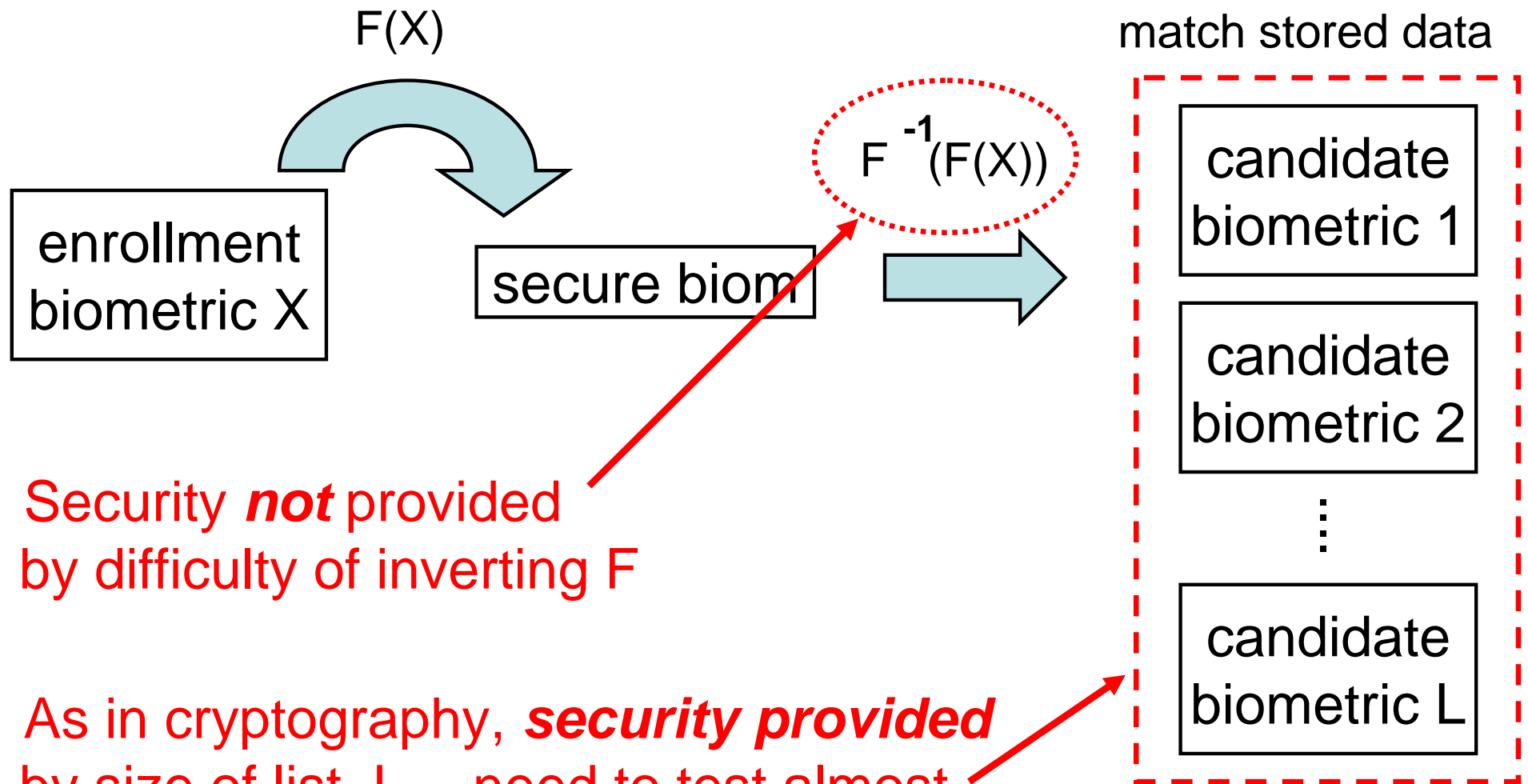


Security = number “missing” bits
= original bits – syndrome bits
 Translates into number guesses to identify original biometric w.h.p.

Robustness = false-rejection rate
 Robustness to variations in biometric readings achieved by syndrome decoding process (syndrome + noisy biometric => original biometric)

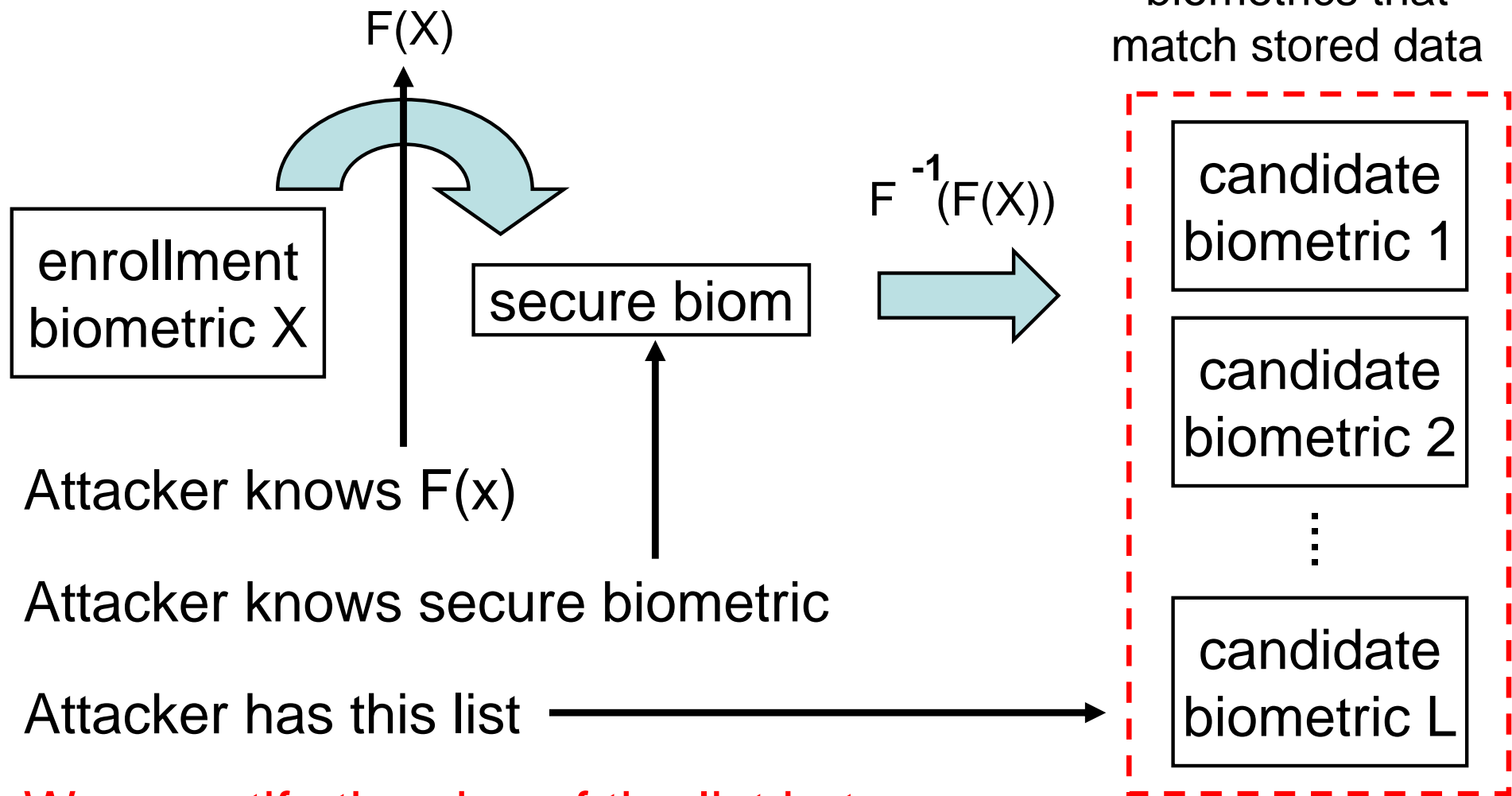
Fewer syndrome bits = greater security, but less robustness

Security Analysis



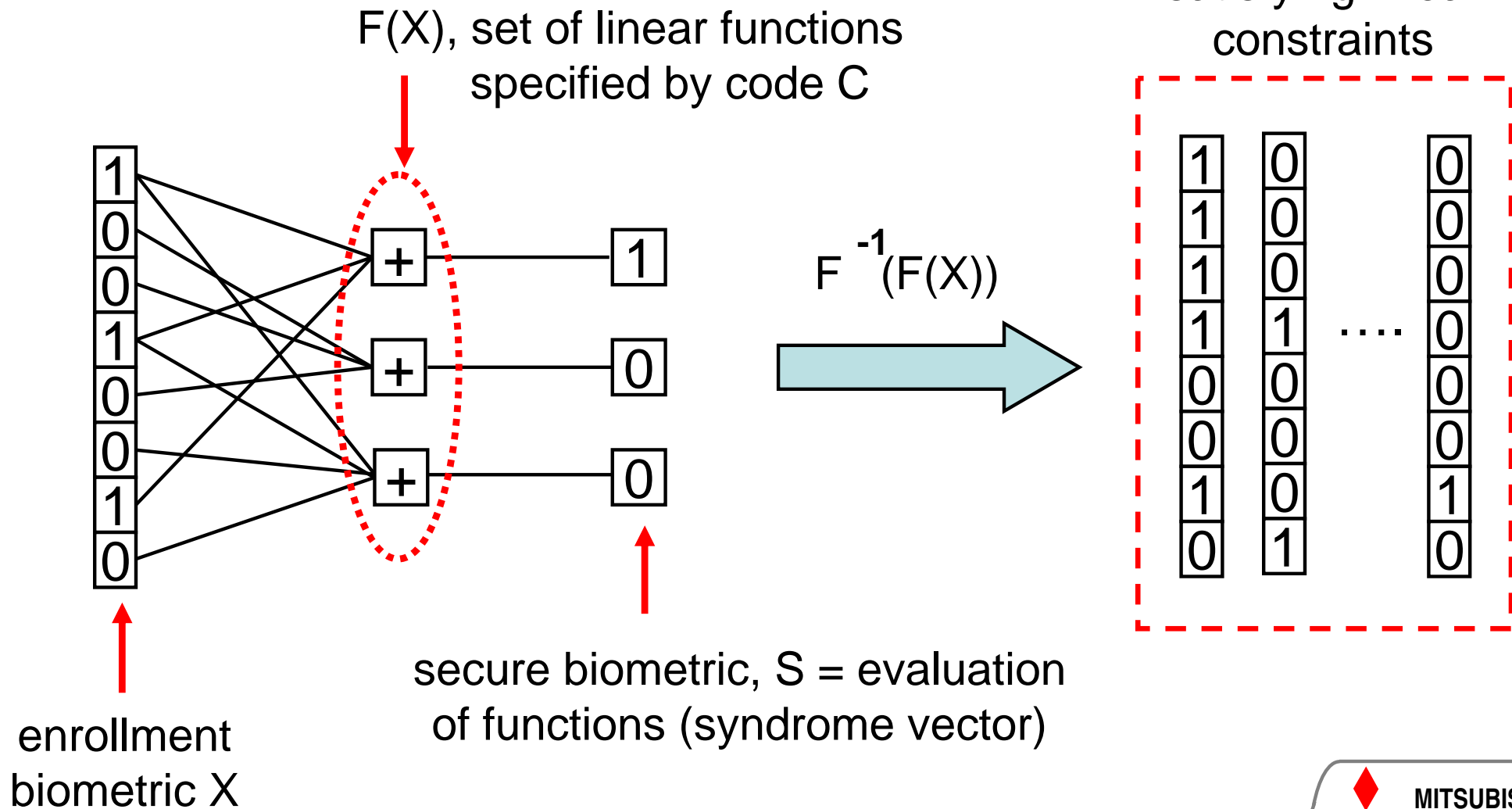
- Security **not** provided by difficulty of inverting F
- As in cryptography, **security provided** by size of list, L -- need to test almost all L to identify enrollment biometric

Quantifying Security



- Attacker knows $F(x)$
- Attacker knows secure biometric
- Attacker has this list
- We quantify the size of the list in terms of measurable characteristics of F

Security of Syndrome-Based System



Security/Robustness evaluation: information-theoretic analysis

X = biometric feature (length n binary vector)

S = syndrome (length nR_{SW} binary vector, R_{SW} is compression rate)

Y = biometric probe (length n binary vector)

Security corresponds to number of missing bits

Guess from typical sequences in bin

$2^{H(X|S)}$ guesses required for successful attack w.h.p.

$$R_{sec} = H(X|S) = H(X, S) - H(S) = H(X) - H(S) = H(X) - nR_{SW}$$

Lower values of $R_{SW} \rightarrow$ higher security

Robustness determined by Slepian-Wolf error exponent

$$\Pr[\text{false rejection}] = \exp\{-n E_{SW}(R_{SW})\}$$

Lower values of $R_{SW} \rightarrow$ higher false-rejection-rate

Security/Robustness range

$R_{SW} < (1/n) H(X)$ needed for positive information security

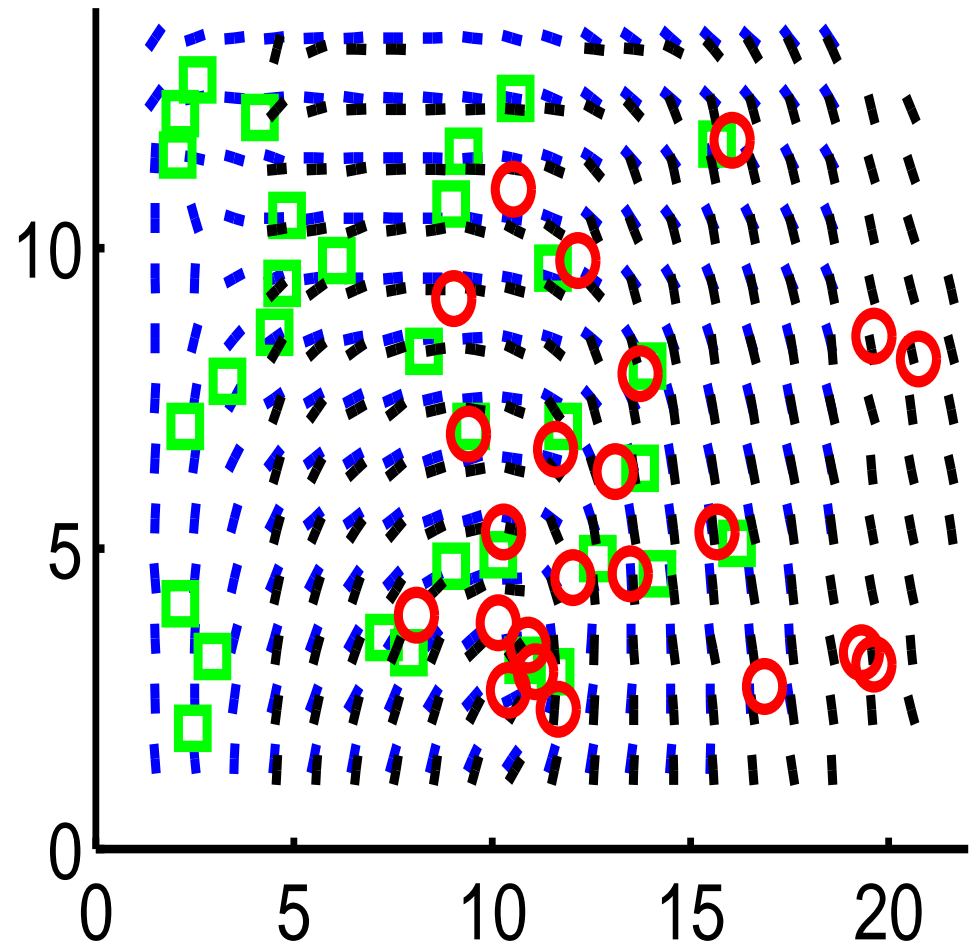
$R_{SW} > (1/n) H(X|Y)$ needed for positive error exponent

Measurement Channel for Fingerprints

Noise is quite different from regular additive noise

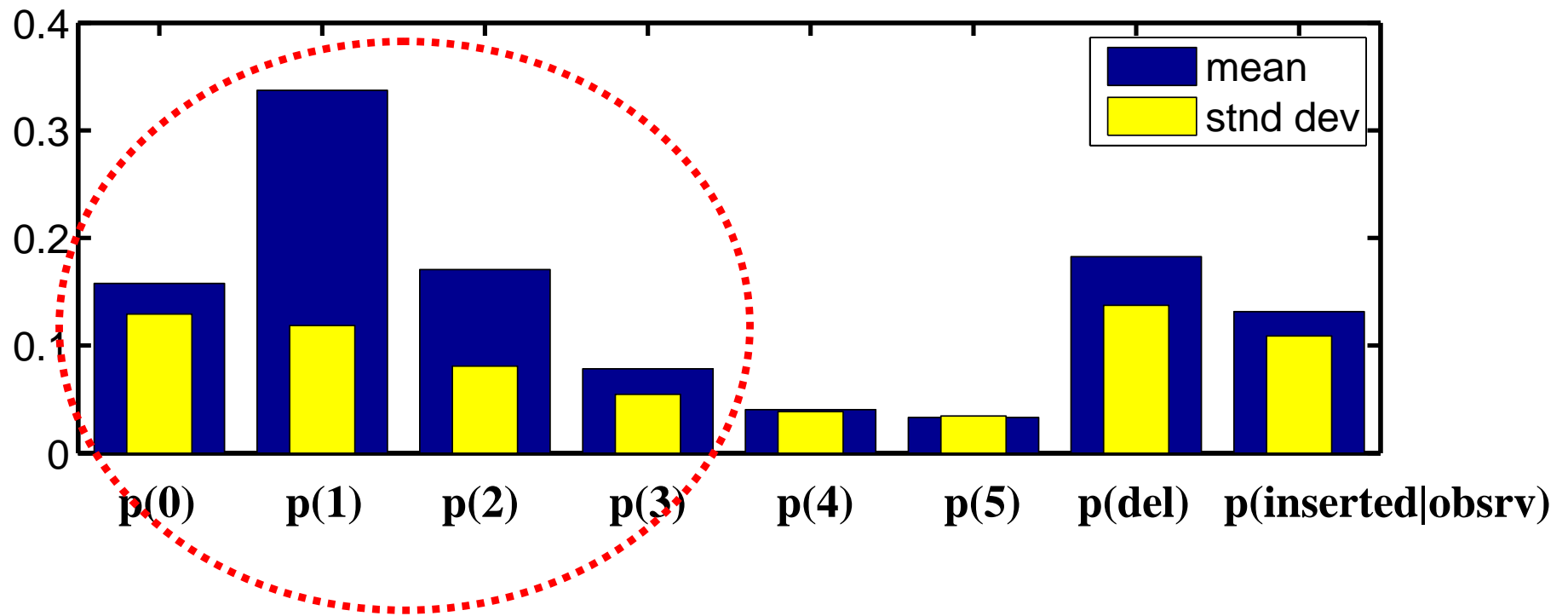
- enrollment minutiae
- probe minutiae

1. minutiae move
2. minutiae disappear
3. minutiae appear
4. enroll/probe not aligned



Movement statistics from Mitsubishi database (1000x15 prints)

Plot distribution of max x- or y- movement, deletions & insertions



Most minutia movements within distance 3

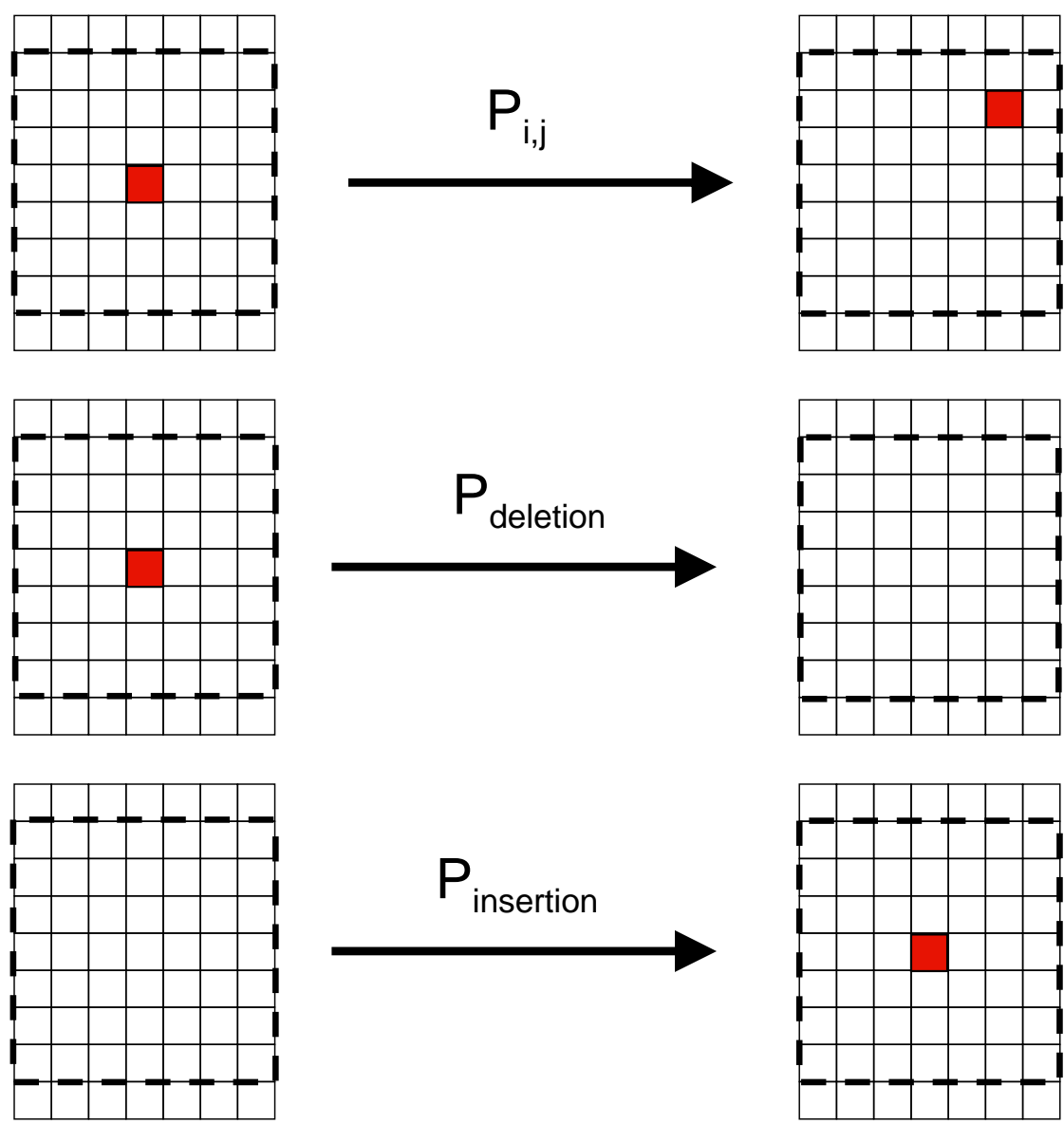
Features of fingerprint biometrics

- Need representation format for fingerprints
 - Fingerprints commonly represented as minutiae (x,y,Θ)
 - Use binary grid to represent minutiae point locations
 - Orientation could be indicated (use Θ rather than 1)



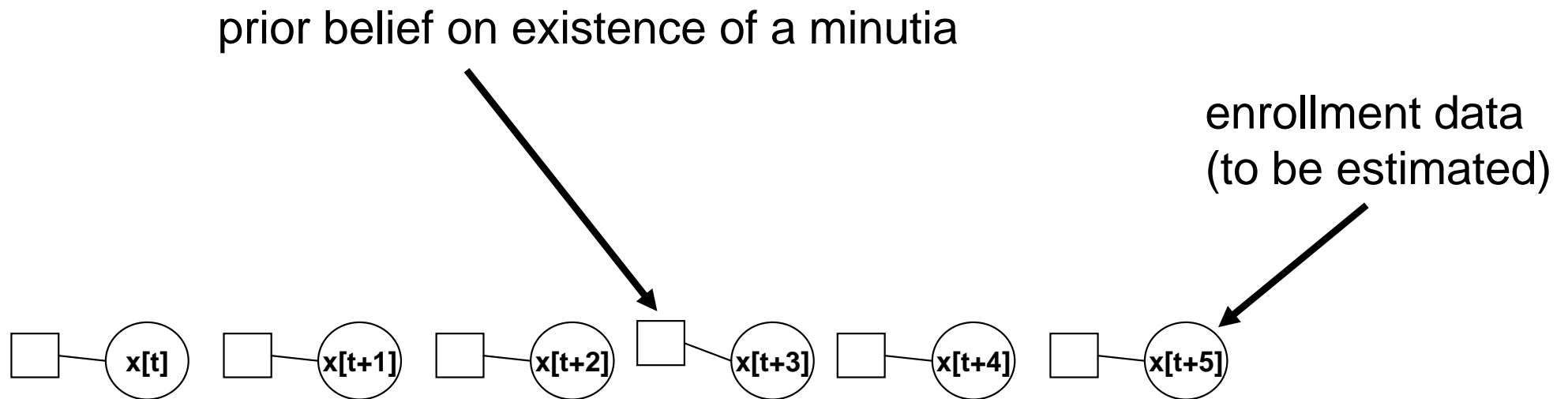
0	0	0	0	1	0	0	0	1	0
0	1	0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	1	0
1	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0
1	0	1	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	1	0	0
0	0	1	0	1	1	0	0	0	0
0	1	1	0	0	1	1	1	0	1
0	0	1	1	1	0	1	1	0	1

Minutiae movement / deletion / insertion model

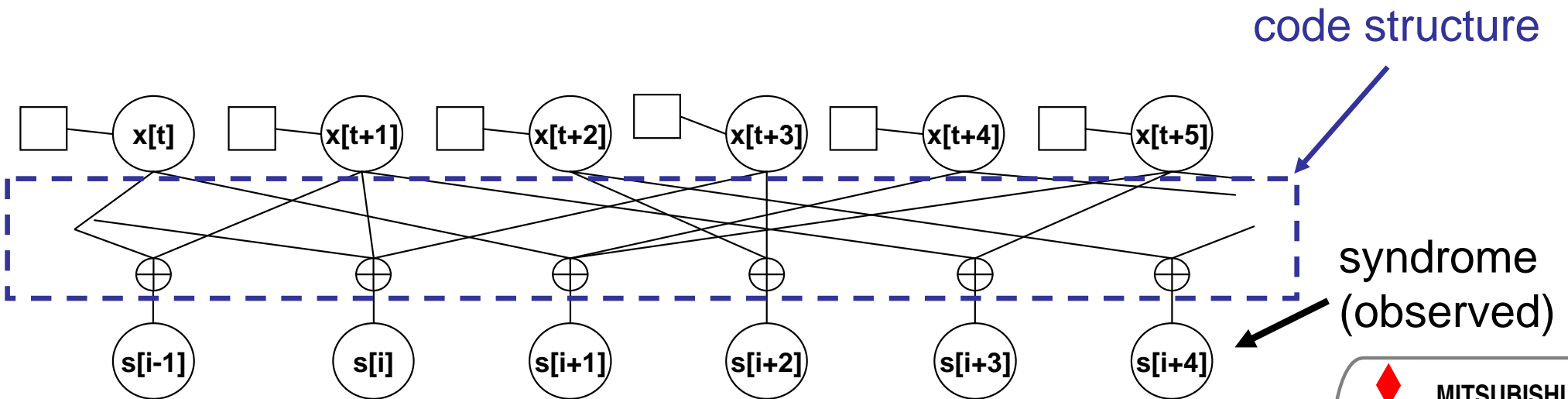


To perform Slepian-Wolf decoding

Start with the enrollment X we want to estimate...



Add code graph relating enrollment to syndrome S...

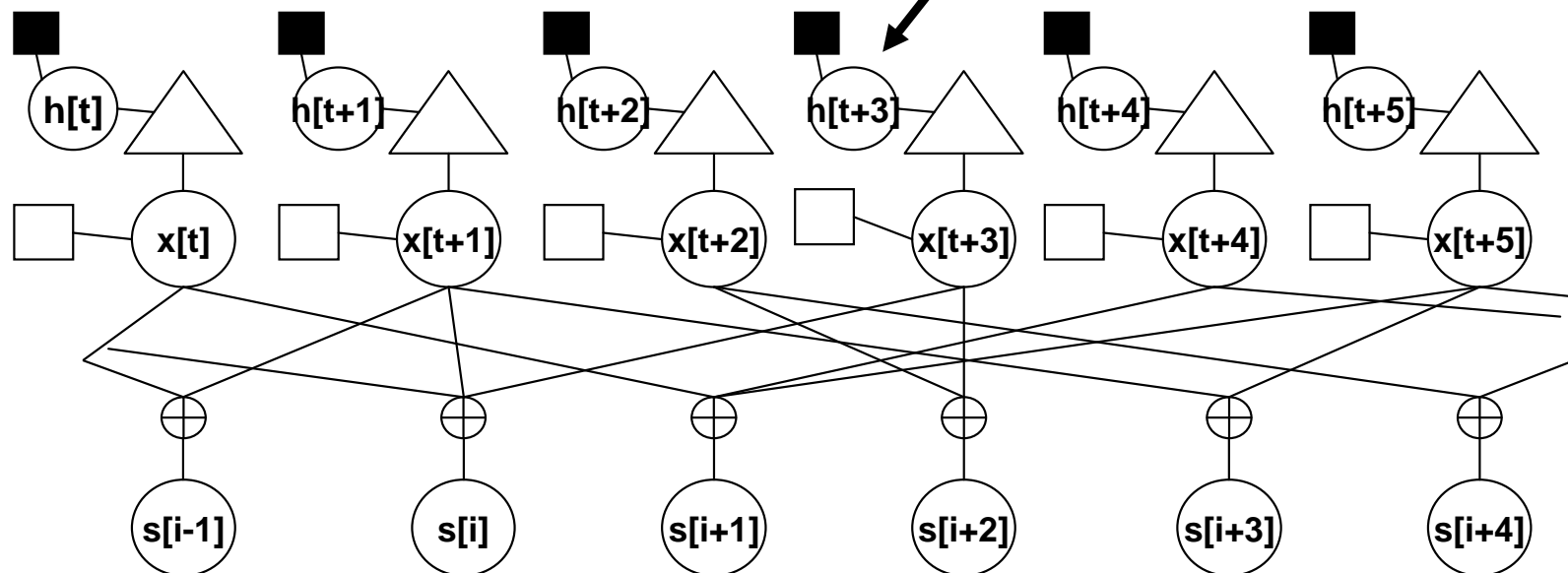


Build up movement model constraints

Minutiae can disappear...

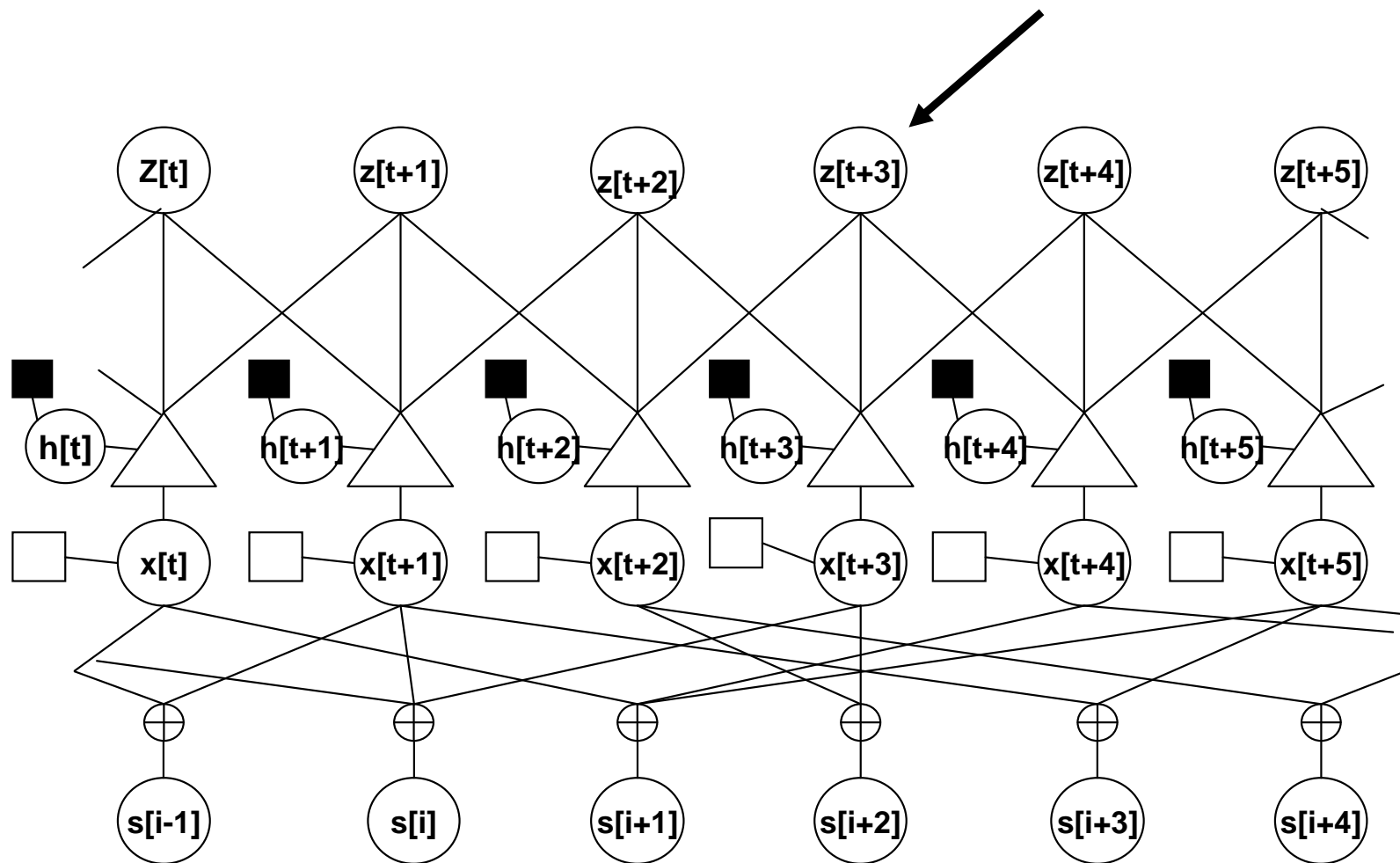
prior belief that a minutia at position $t+2$ is deleted

variable denoting whether a minutia at position $t+3$ is deleted



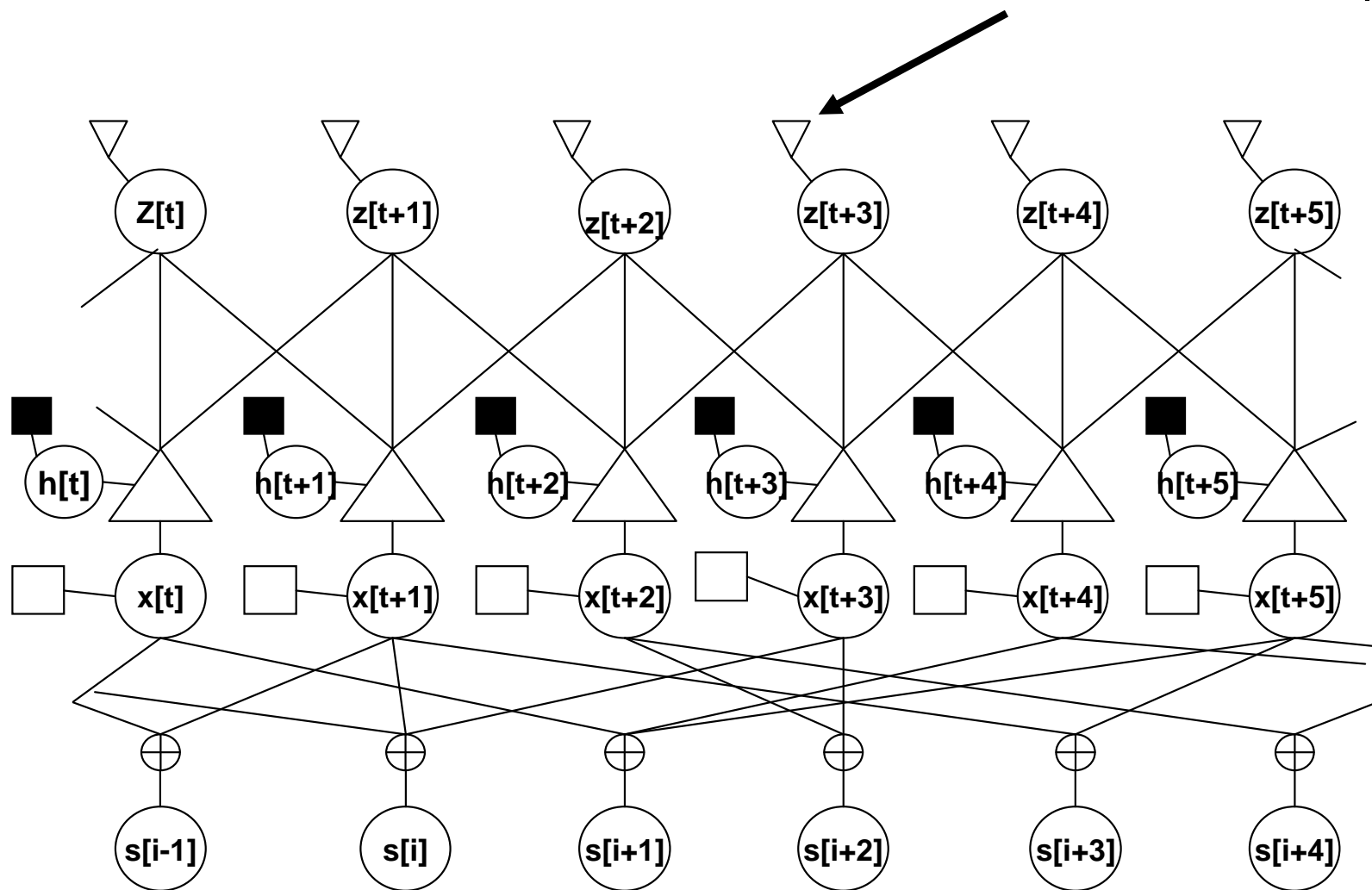
Minutiae can move...

variable indicating where an observed minutia at position $t+3$ came from

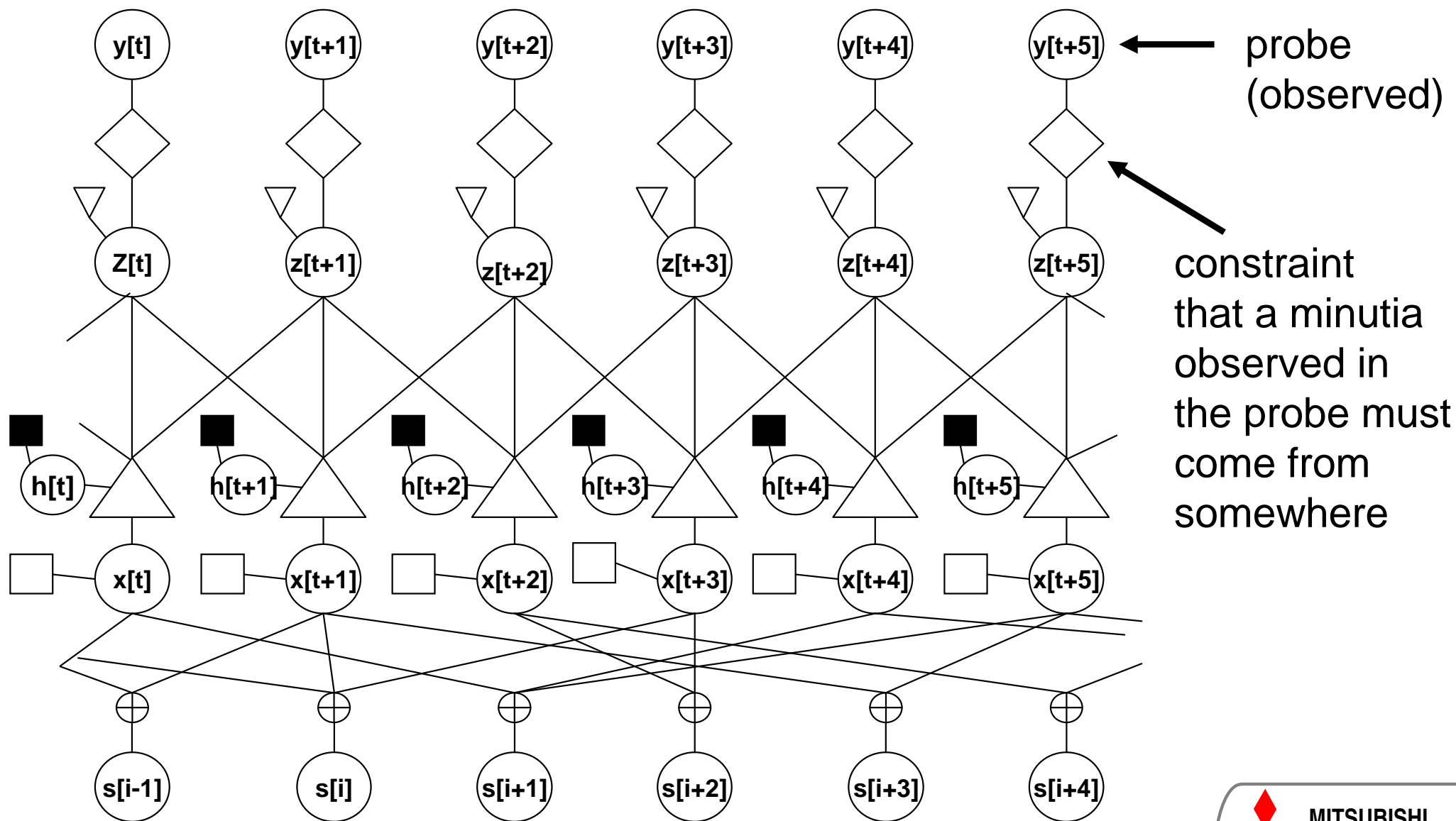


Minutiae can appear...

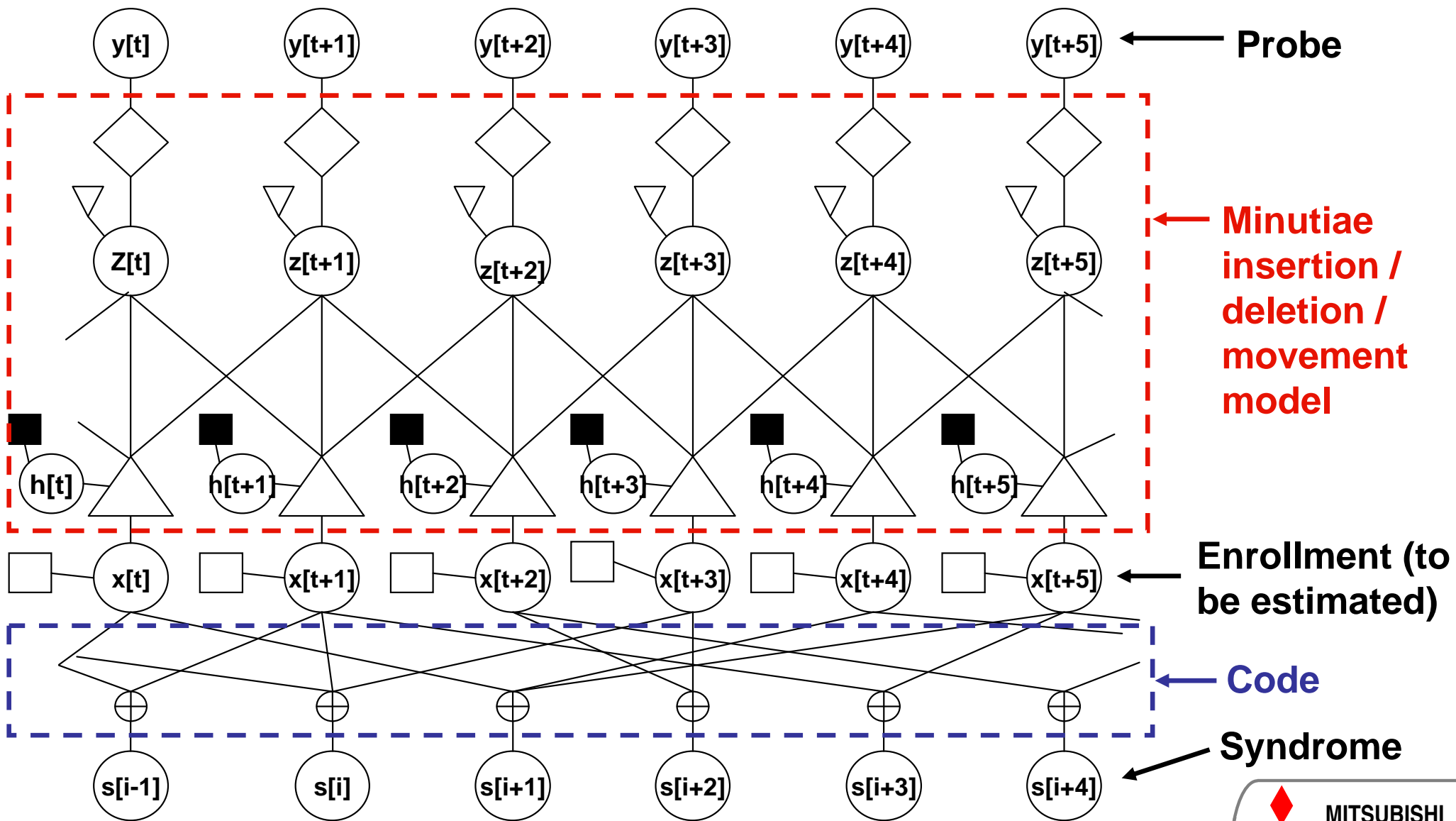
prior belief that a spurious
minutia is inserted at position $t+3$



And, minutiae are observed in Y (noisy version)

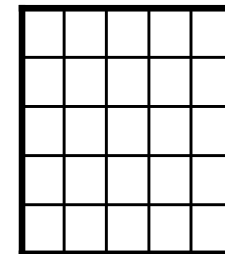
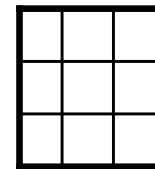


All together... only given probe & syndrome



Experimental results – synthetic data

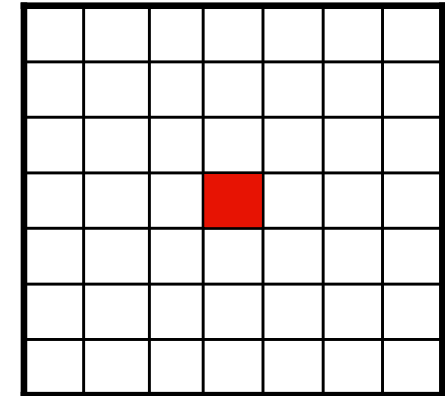
- False reject rates (FRR) depends on how far minutiae can move, e.g., at $R_{sw} = 0.06$
 - Minutiae move +/- 1: FRR < 1%
 - Minutiae move +/- 2: FRR \approx 4.5%



SW code rate	Successful attack rate zero-probe	Simple movement model FRR	Full model with Pr[erasure] = 0, FRR	Full model with Pr[insertion] = 0, FRR	Full model with erasures and insertions, FRR
0.08	0.57	0.48	N/A	N/A	N/A
0.07	0.33	0.76	N/A	N/A	N/A
0.06	0	1.0	0.70e-2	0.94e-2	4.6e-2
0.05	0	1.0	0.42e-2	1.7e-2	7.9e-2

Experimental results – real data

- Model constrains minutia movement to ± 3
- Encoded with fixed rate 0.94 LDPC code
- **Average FRR = $15.8e-2$**
- **Average FAR = $0.47e-2$**



# enrolled minutiae H(x)	Number test enrollments	Slepian-Wolf Rate	FRR	FRR (# probes)	FAR	FAR (# probes)
31 (0.04100)	195	0.04102	$11.6e-2$	2736	$0.98e-2$	11e4
32 (0.04211)	139	0.04128	$13.3e-2$	1944	$0.33e-2$	7.8e4
33 (0.04322)	107	0.04189	$14.9e-2$	1506	$0.24e-2$	6.0e4
34 (0.04432)	79	0.04275	$20.2e-2$	1101	$0.11e-2$	4.4e4
35 (0.04541)	59	0.04309	$32.3e-2$	824	$0.03e-2$	3.3e4

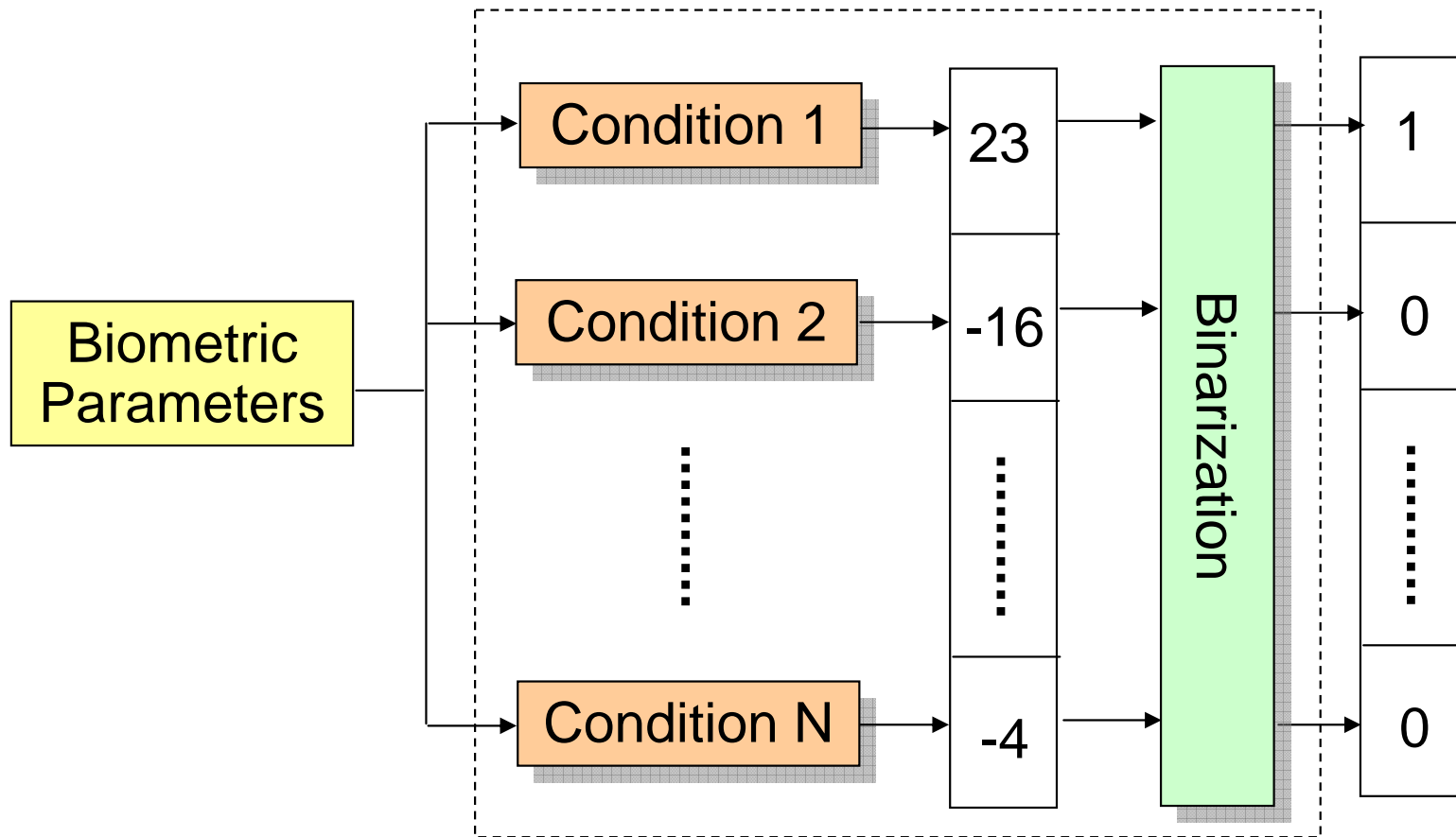
Observations

- The work so far [*Draper, et al: ICASSP 2007*]
 - Used binary grid representation of minutiae points in fingerprint
 - Syndrome coding on binary string from grid representation
 - Required statistical model of the movement, insertion, and deletions intrinsic to minutiae as part of decoding
- Problems
 - Representation is sparse and difficult to model
 - Statistics of binary string are not well suited for existing codes
 - Improved performance might be attainable by designing better codes that account for biased nature of source and asymmetry of measurement channel -- complicated process

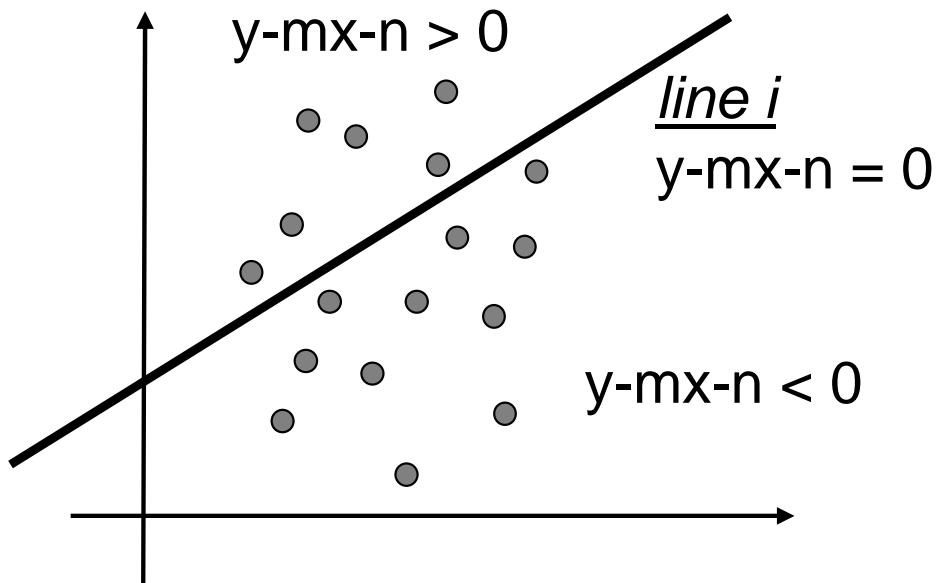
Our Current Direction: Apply Pre-processing

- Basic idea
 - Pre-process fingerprint data to produce a binary string that is statistically well suited for existing codes
 - Then, perform syndrome coding on resulting binary string
- Desired statistical properties
 - Each bit in string should be 50% zero and 50% one
 - Different bits in the same string should be independent
 - Bit strings for different users should be independent
 - Bit strings for different samples of same user should be statistically dependent

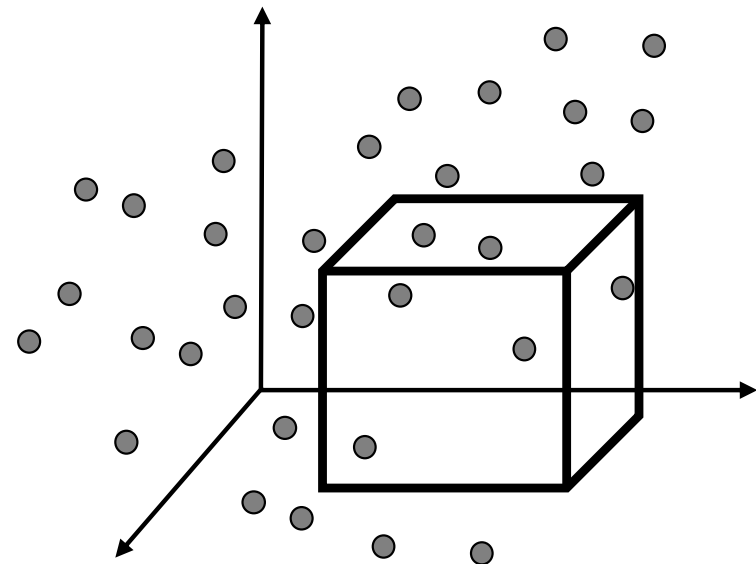
Pre-processing Operations



Various Types of Conditions - Random Lines / Cubes

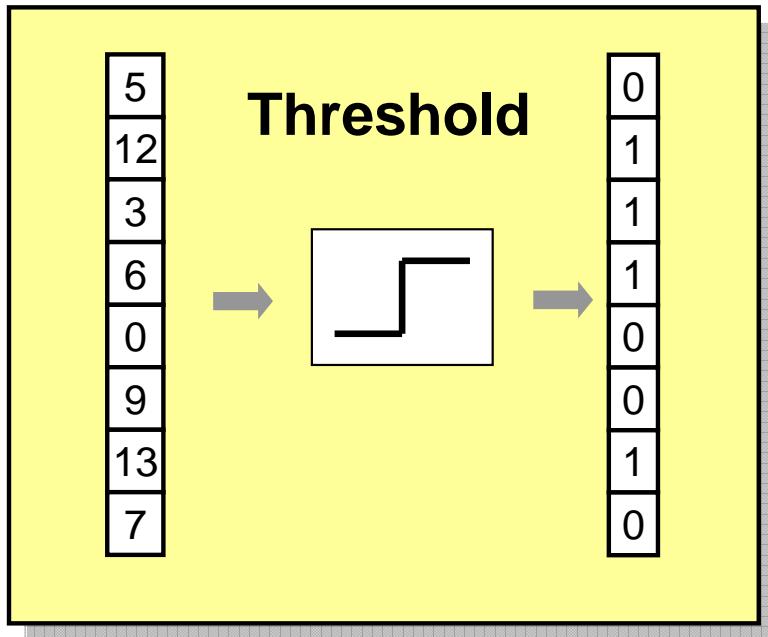


- Randomly generated set of lines
- Subtract number of minutiae above and below each line
- Vector values are uniform on $[-M, M]$ where M is the # of minutiae points
- Correlated components when lines are very similar

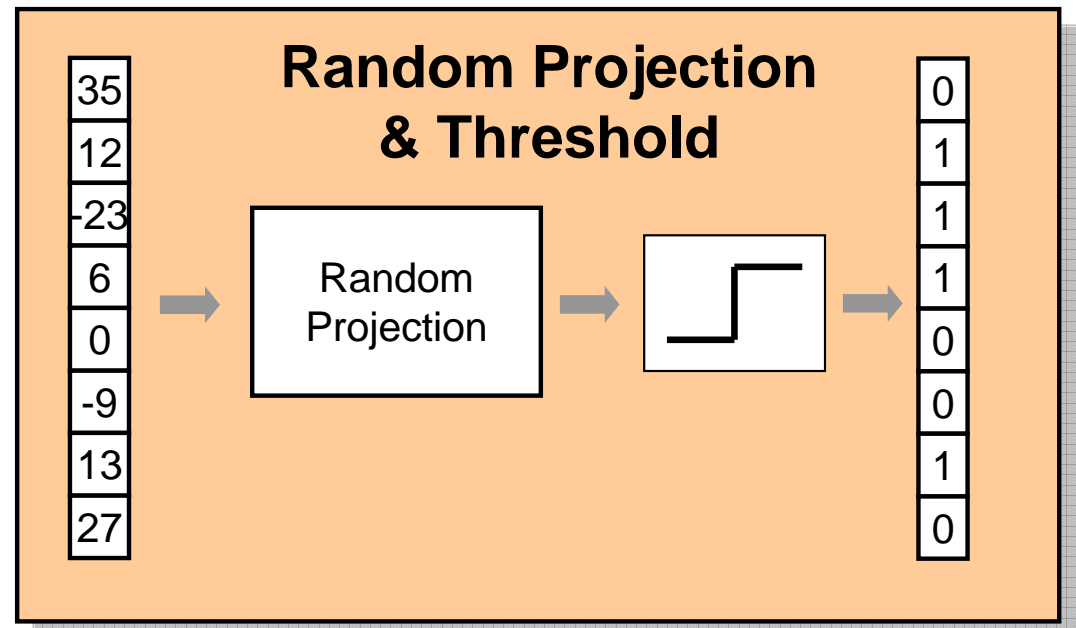


- Consider total minutiae in a cube
- Cubes make it possible to incorporate minutiae orientation
- Overlap and volume of cubes needs careful consideration

Various Types of Binarization Schemes



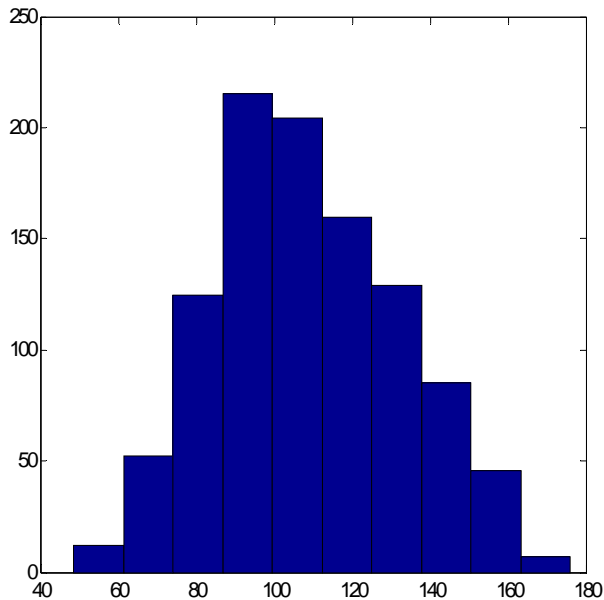
Simplest approach



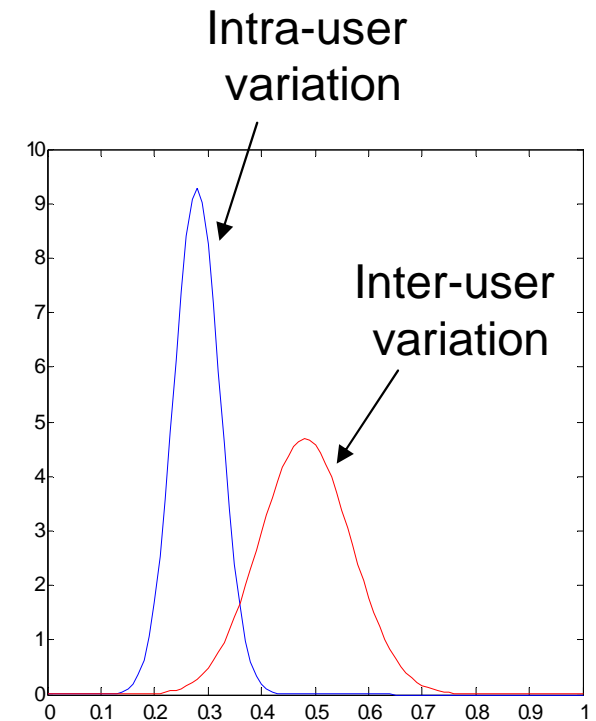
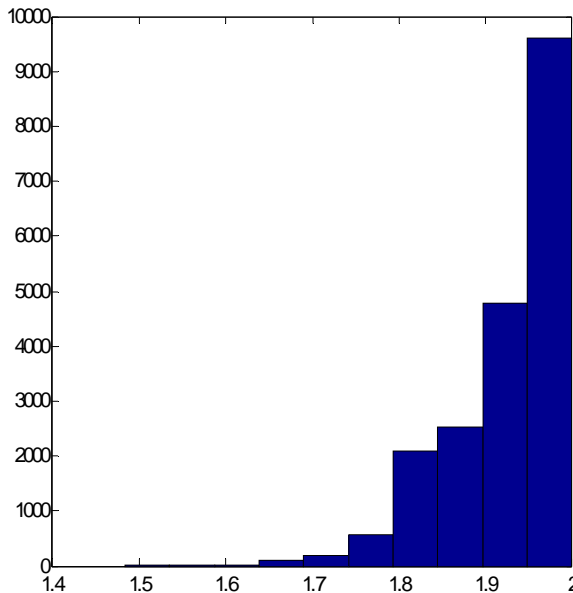
Removes correlation in integer vector
Improves separation between users

Sample Results

Average Number of 1's



Pair-wise Entropy



- Randomly generated cubes – fixed for all users
- # of minutiae points in each cube determined for each user
- Component-wise thresholding wrt median of that component (calculated over all users)

Concluding Remarks

- Work on distributed video coding will inspire new applications and uses beyond compression
 - Secure biometrics is an exemplary case
 - Image authentication is another [*Y.C. Lin, et al., Stanford*]
- Necessary to apply codes that match channel properties
 - Design new codes for given channel / measurement model
 - Transform the input data to match the capabilities of the code and assumed channel
- Our pre-processing work is still in an early stage
 - Currently focused on achieving desired statistical properties
 - Plan to evaluate performance in syndrome coding framework very soon
 - Stay tuned...