

How to Generate Keys from Facial Images and Keep your Privacy at the Same Time

Margarita Osadchy
University of Haifa

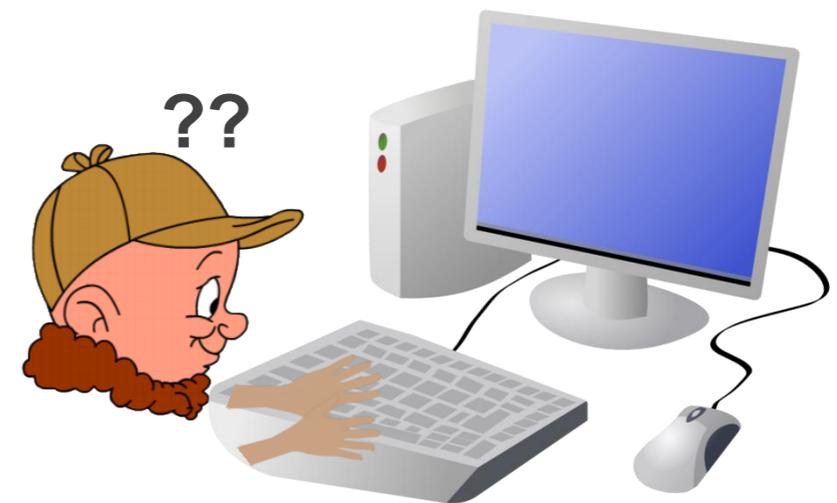
Joint work with Mahmood Sharif and Orr Dunkelman

Motivation

- ❖ Key-Derivation: generating a secret key, from information possessed by the user
- ❖ Passwords are the most widely used means for key derivation,
- ❖ but...

Motivation

- ❖ Passwords are:
 1. Forgettable



Motivation

❖ Passwords are:

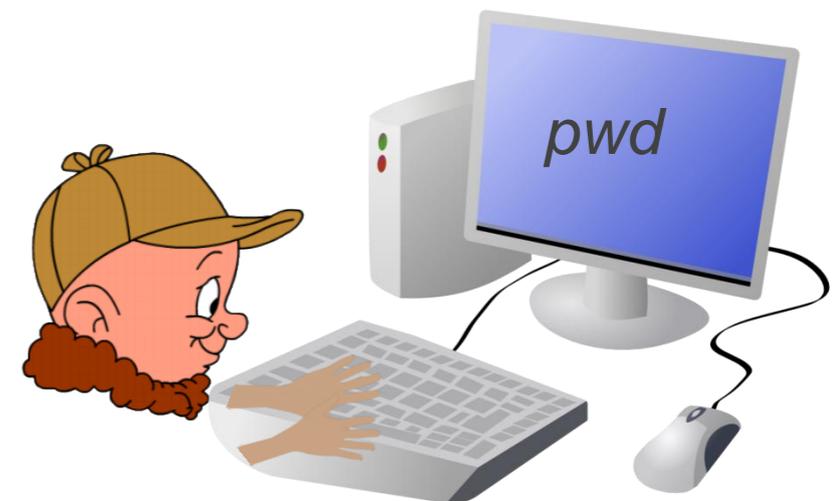
1. Forgettable
2. Easily observable (shoulder-surfing)



Motivation

❖ Passwords are:

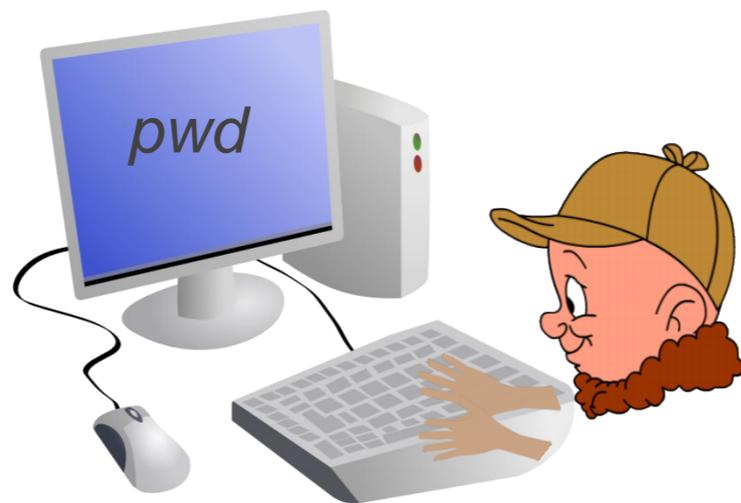
1. Forgettable
2. Easily observable (shoulder-surfing)
3. Low entropy



Motivation

❖ Passwords are:

1. Forgettable
2. Easily observable (shoulder-surfing)
3. Low entropy
4. Carried over between systems



Motivation

- ❖ Suggestion: use biometric data for key generation
- ❖ Problems :
 1. It is hard/impossible to replace the biometric template in case it gets compromised
 2. Privacy of the users



Privacy of Biometric Data



X



K



The Fuzziness Problem

Two images of the same face are rarely identical (due to lighting, pose, expression changes)

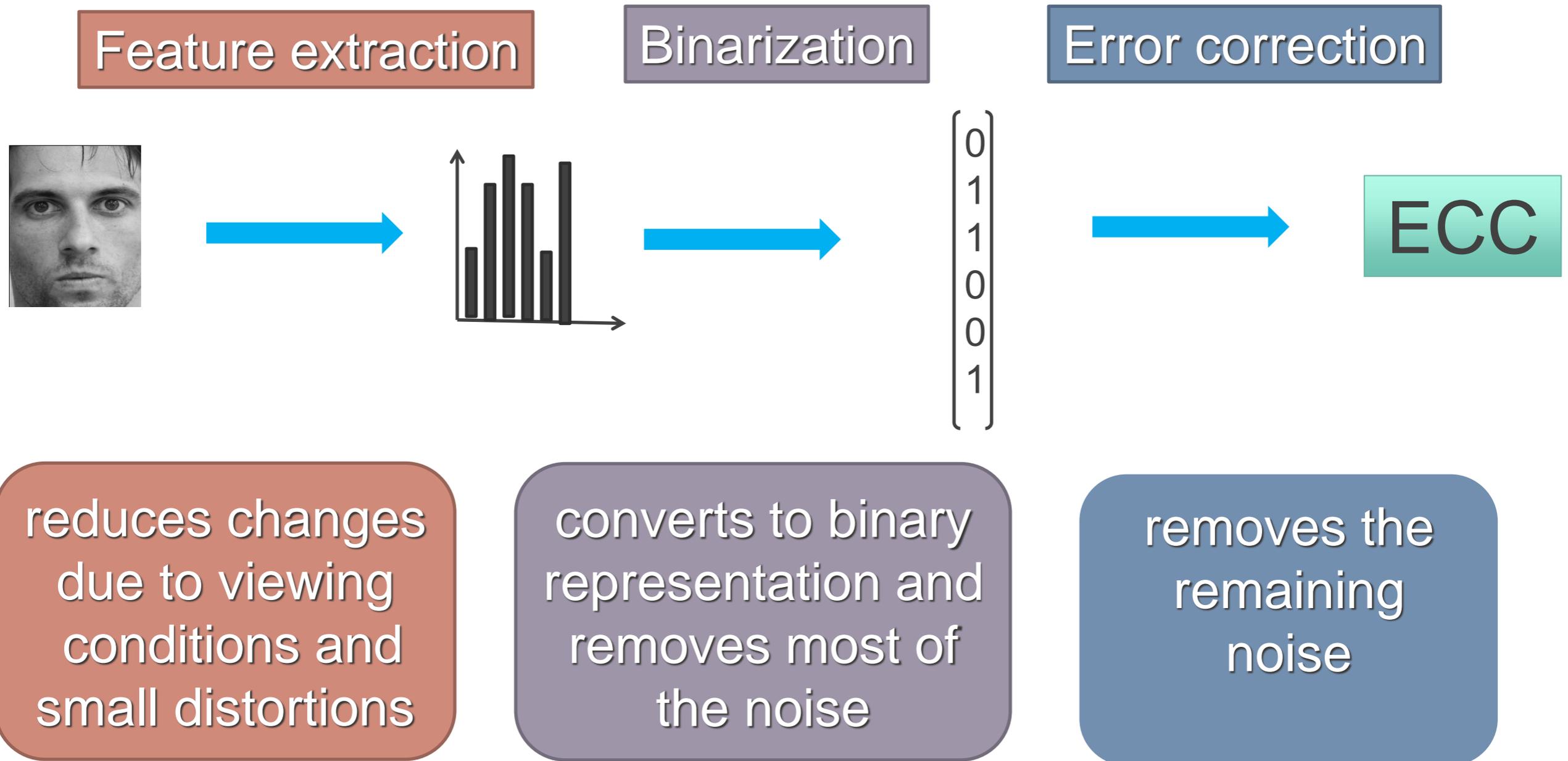


- **Taken one after the other**
- **81689 pixels are different**
- **only 3061 pixels have identical values!**

The Fuzziness Problem

- ❖ Two images of the same face are rarely identical (due to lighting, pose, expression changes)
- ❖ Yet we want to consistently create the same key for the user every time
- ❖ The fuzziness in the samples is handled by:
 1. Feature extraction
 2. Use of error-correction codes and helper data

3-Step Noise Reduction Process



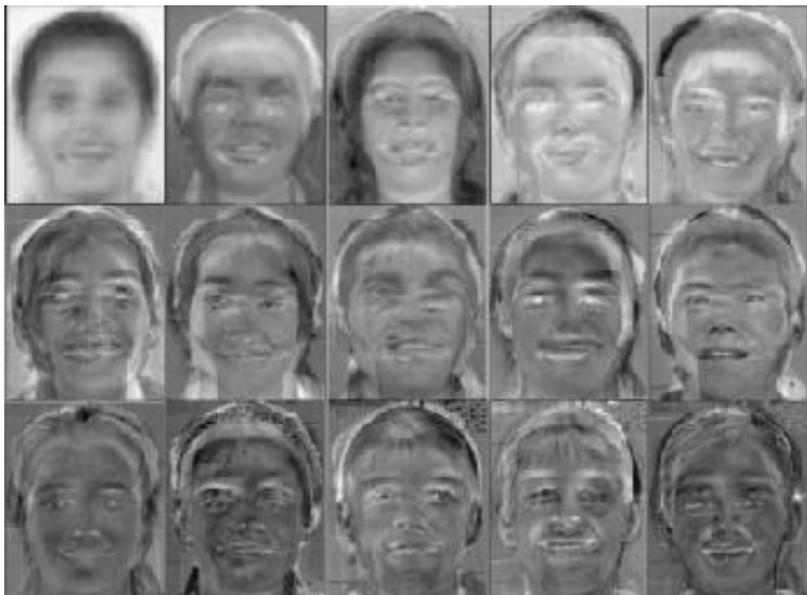
Feature Extraction

User-specific features:

Eigenfaces (PCA)

Fisherfaces (FLD)

training step produces user specific parameters, stored for feature extraction



Generic Features

Histograms of low-level features, e.g.: LBPs, SIFT

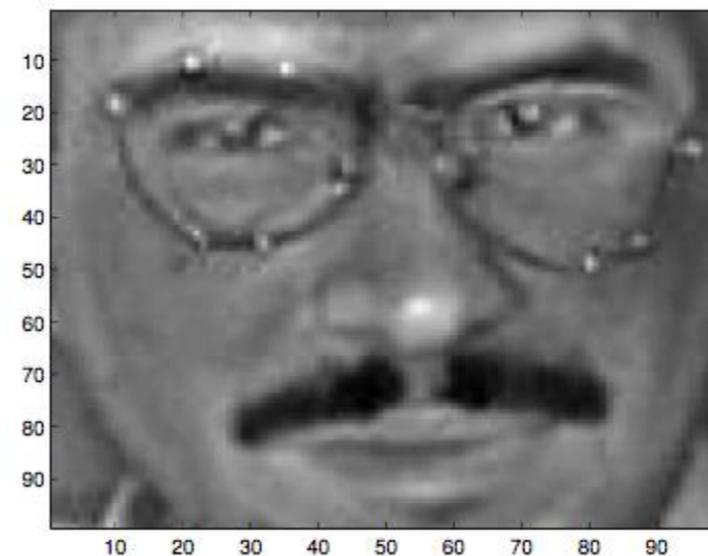
Filters : Gabor features, etc

No training, no user specific information is required

Feature Extraction

Previous Work

- ❖ [FYJ10] used Fisherfaces - public data looks like the users:



- ❖ Very discriminative (better recognition)
- ❖ But compromises privacy.

Can't be used!

Use Generic Features?

- ❖ Yes, but require caution.
- ❖ In [KSVAZ05] high-order dependencies between different channels of the Gabor transform
- ❖ → correlations between the bits of the suggested representation

Binarization

❖ Essential for using the cryptographic constructions

❖ ~~Some claim: non invertibility [TGN06]~~

Biometric features can be approximated

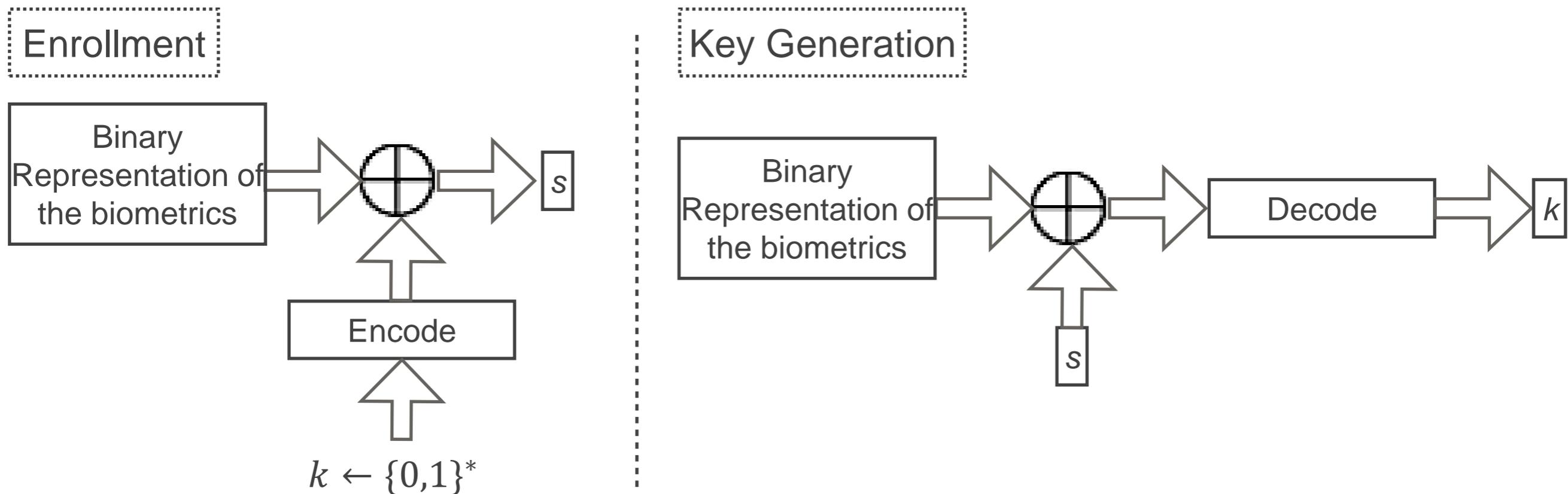
❖ By :

- Sign of projection
- Quantization

Quantization is more accurate, but requires storing additional private information.

Cryptographic Noise Tolerant Constructions

❖ Fuzzy Commitment [JW99]:



❖ Other constructions: Fuzzy Vault [JS06], Fuzzy Extractors [DORS08]

Previous Work

Problems

1. Short keys
2. Non-uniformly distributed binary strings as an input for the fuzzy commitment scheme
3. Dependency between bits of the biometric samples
4. Auxiliary data leaks personal information
5. No protection when the adversary gets hold of the cryptographic key

Security Requirements

1. Consistency: identify a person as himself (low FRR)
2. Discrimination: impostor cannot impersonate an enrolled user (low FAR)

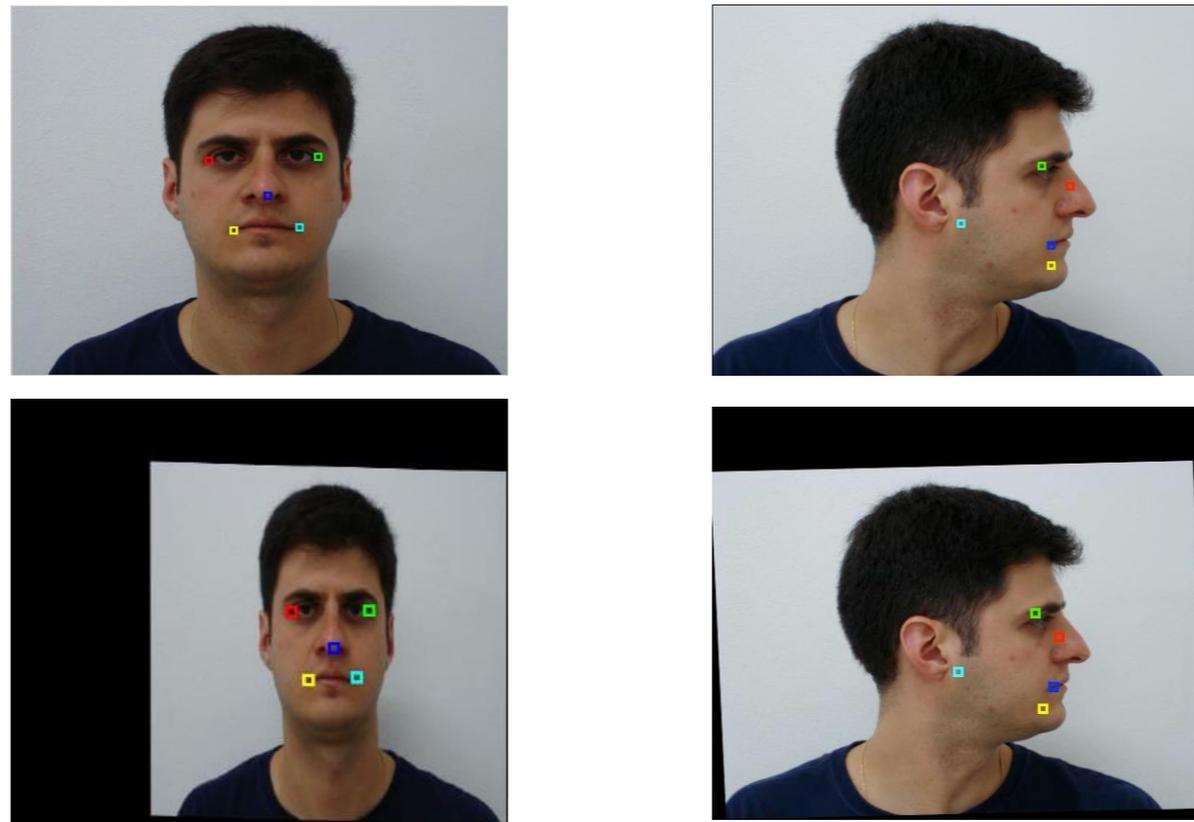
[BKR08]:

3. Weak Biometric Privacy (REQ-WBP): computationally infeasible to learn the biometric information given the helper data
4. Strong Biometric Privacy (REQ-SBP): computationally infeasible to learn the biometric information given the helper data and the key
5. Key Randomness (REQ-KR): given access to the helper data, the key should be computationally indistinguishable from random

Feature Extraction

1. Landmark Localization and Alignment

- ❖ Face landmark localization [ZR12] and affine transformation to a canonical pose:

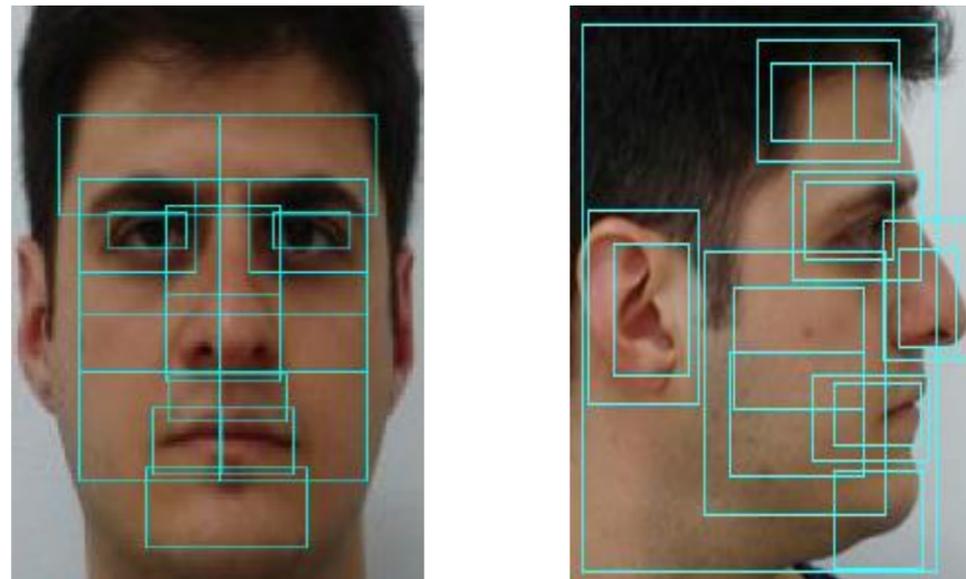


- ❖ An essential step, due to the inability to perform alignment between enrolled and newly presented template

Feature Extraction

2. Feature Extraction

- ❖ Local Binary Patterns (LBPs) descriptors are computed from 21 regions defined on the face:



- ❖ The same is done with Scale Invariant Feature Transform (SIFT) descriptors
- ❖ Histograms of Oriented Gradients (HoGs) are computed on the whole face

Binarization by Projection



x

$$\mathbb{R}^{n_1} \rightarrow \{0, 1\}^{n_2}$$



$$h(x) = \frac{1}{2} (\text{sign}(W^T x) + 1)$$

Binarization by Projection

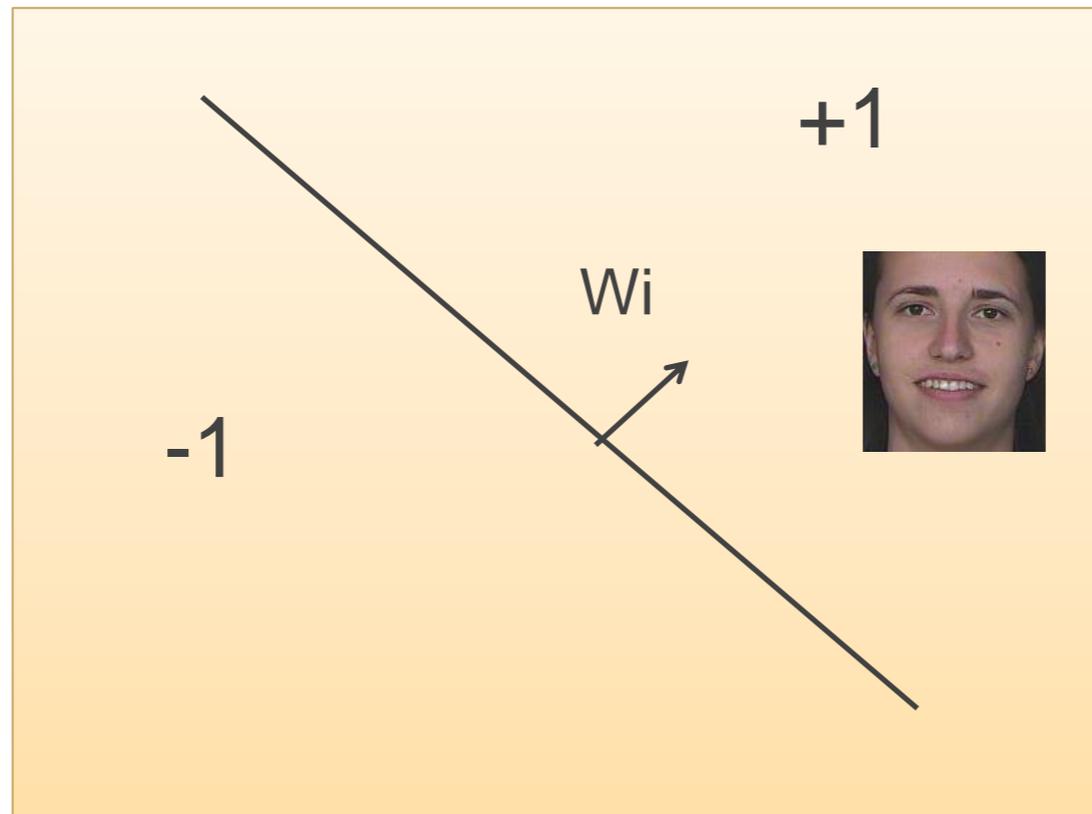


x

$$\mathbb{R}^{n_1} \rightarrow \{0, 1\}^{n_2}$$



$$h(x) = \frac{1}{2} (\text{sign}(W^T x) + 1)$$



$$h_i(x) = 1$$

Binarization by Projection

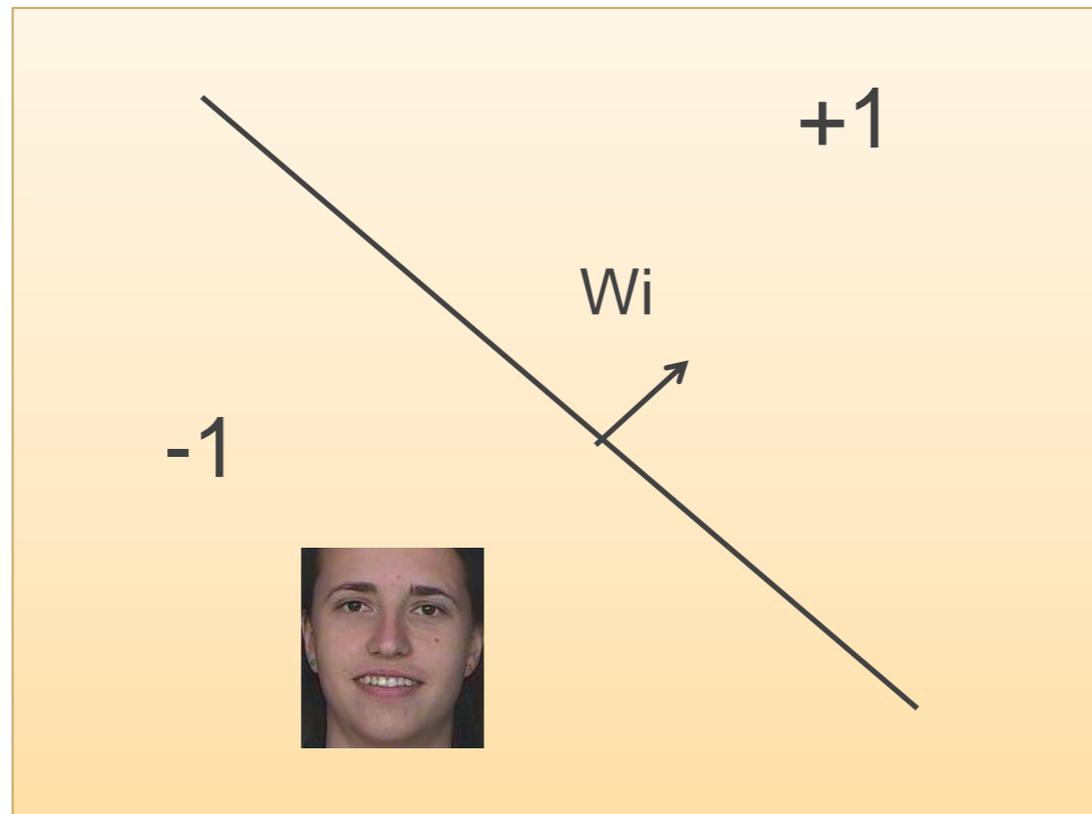


x

$$\mathbb{R}^{n_1} \rightarrow \{0, 1\}^{n_2}$$



$$h(x) = \frac{1}{2} (\text{sign}(W^T x) + 1)$$



$$h_i(x) = 0$$

Binarization by Projection

$$\mathbb{R}^{n_1} \rightarrow \{0, 1\}^{n_2}$$



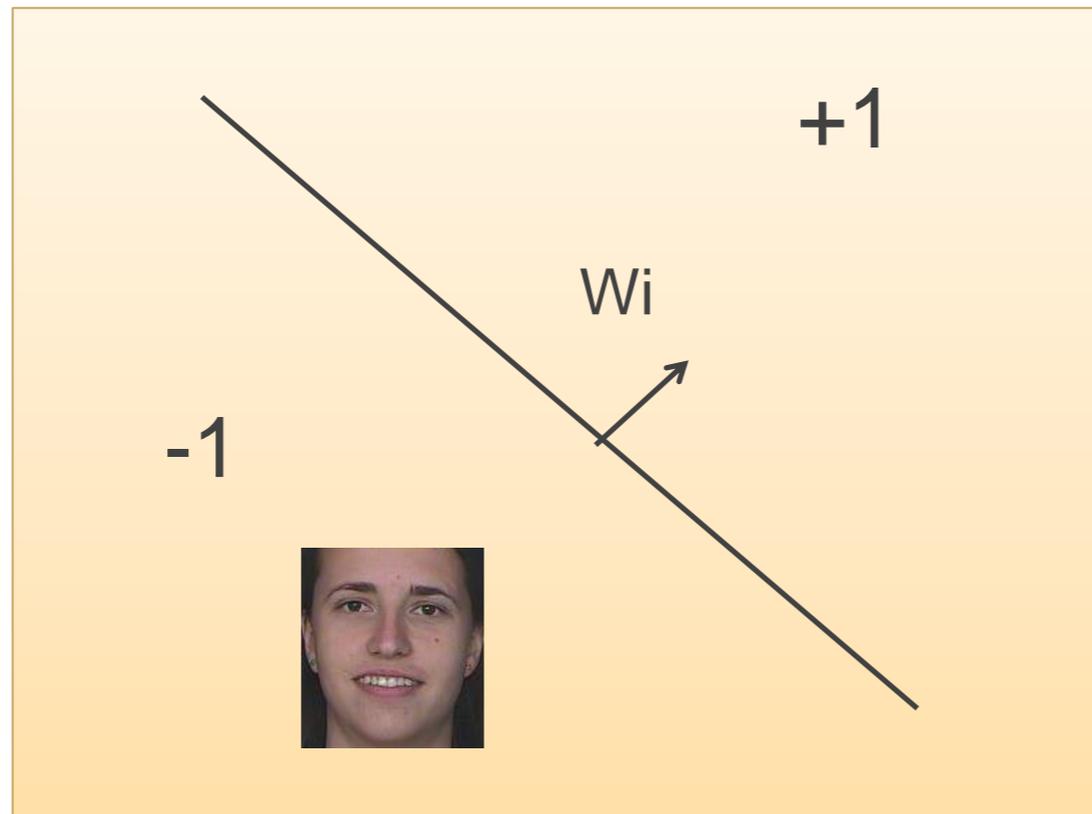
x



$$h(x) = \frac{1}{2} (\text{sign}(W^T x) + 1)$$



$h(x)$?



$$h_i(x) = 0$$

Binarization by Projection

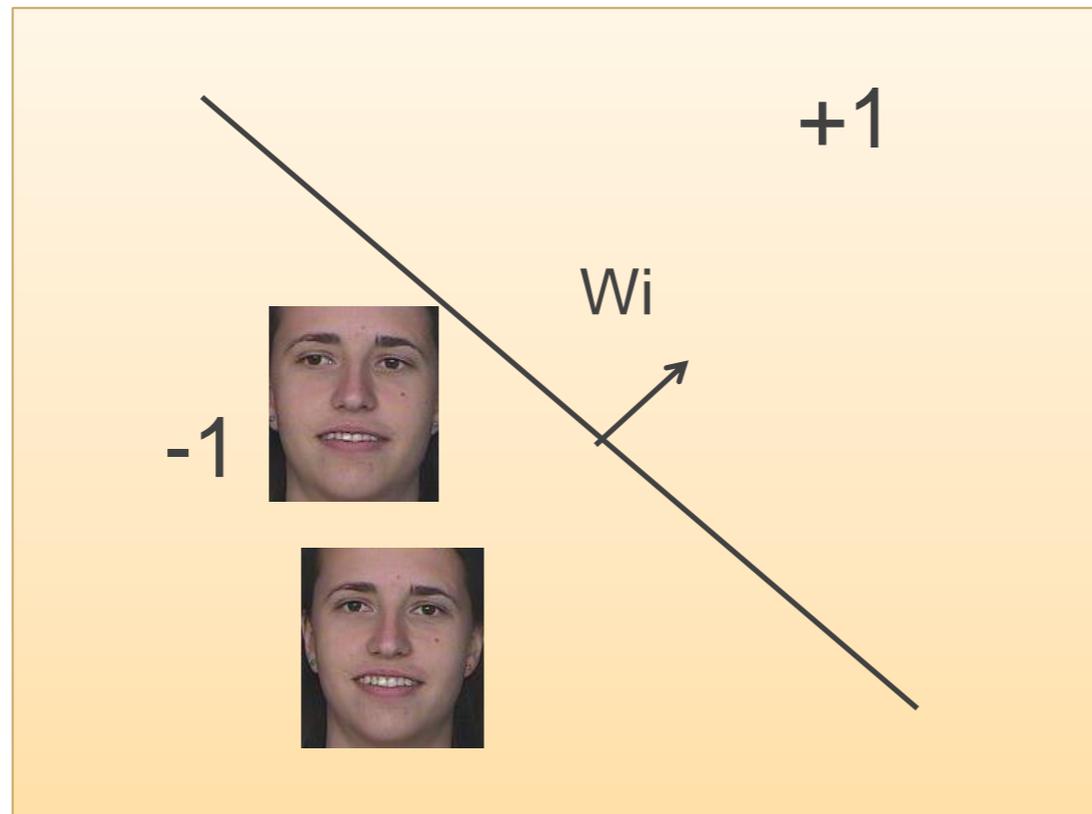


x

$$\mathbb{R}^{n_1} \rightarrow \{0, 1\}^{n_2}$$



$$h(x) = \frac{1}{2} (\text{sign}(W^T x) + 1)$$



$$h_i(x) = 0$$

$$h_i(x') = 0$$

Binarization by Projection

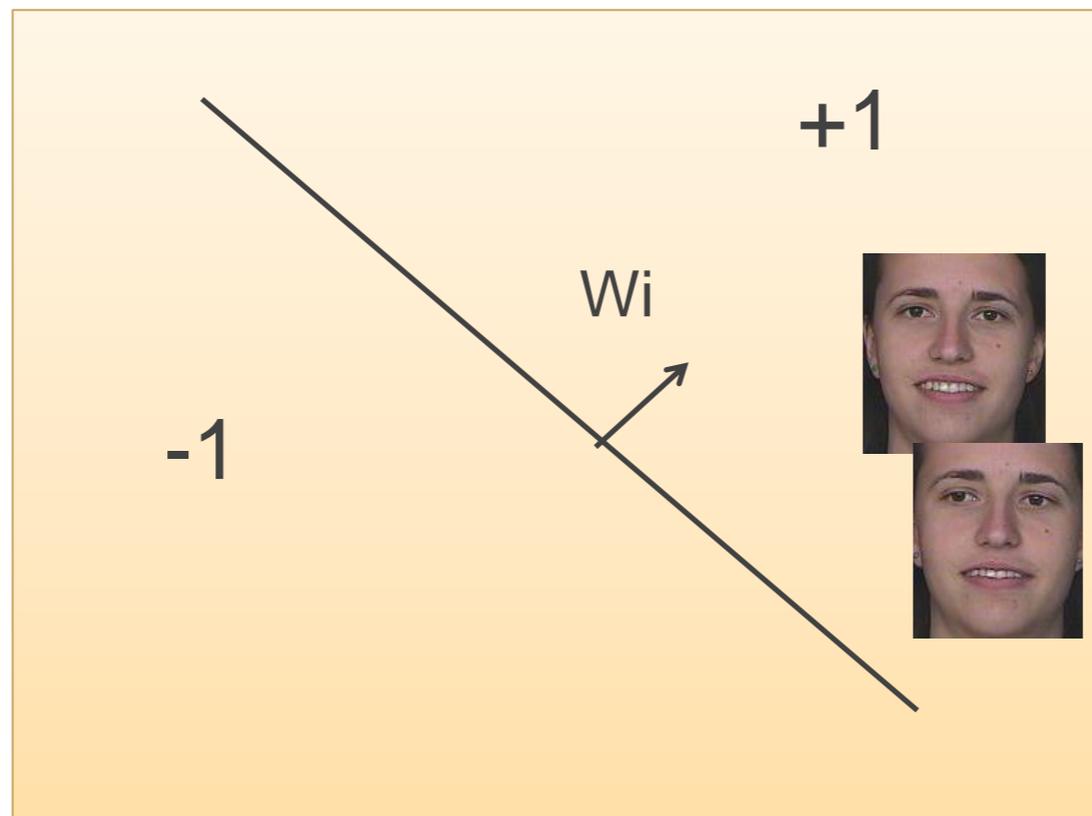


x

$$\mathbb{R}^{n_1} \rightarrow \{0, 1\}^{n_2}$$



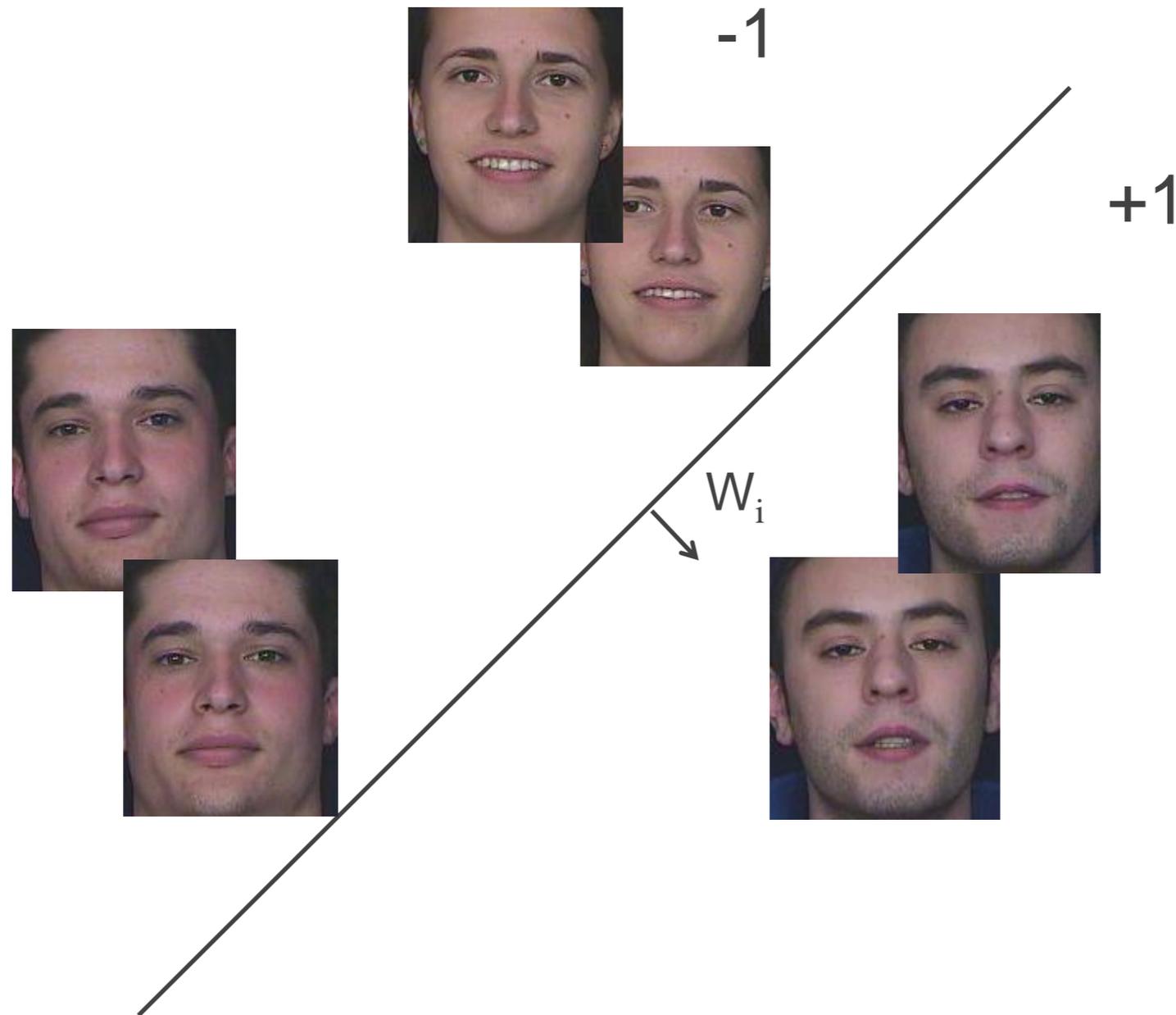
$$h(x) = \frac{1}{2} (\text{sign}(W^T x) + 1)$$



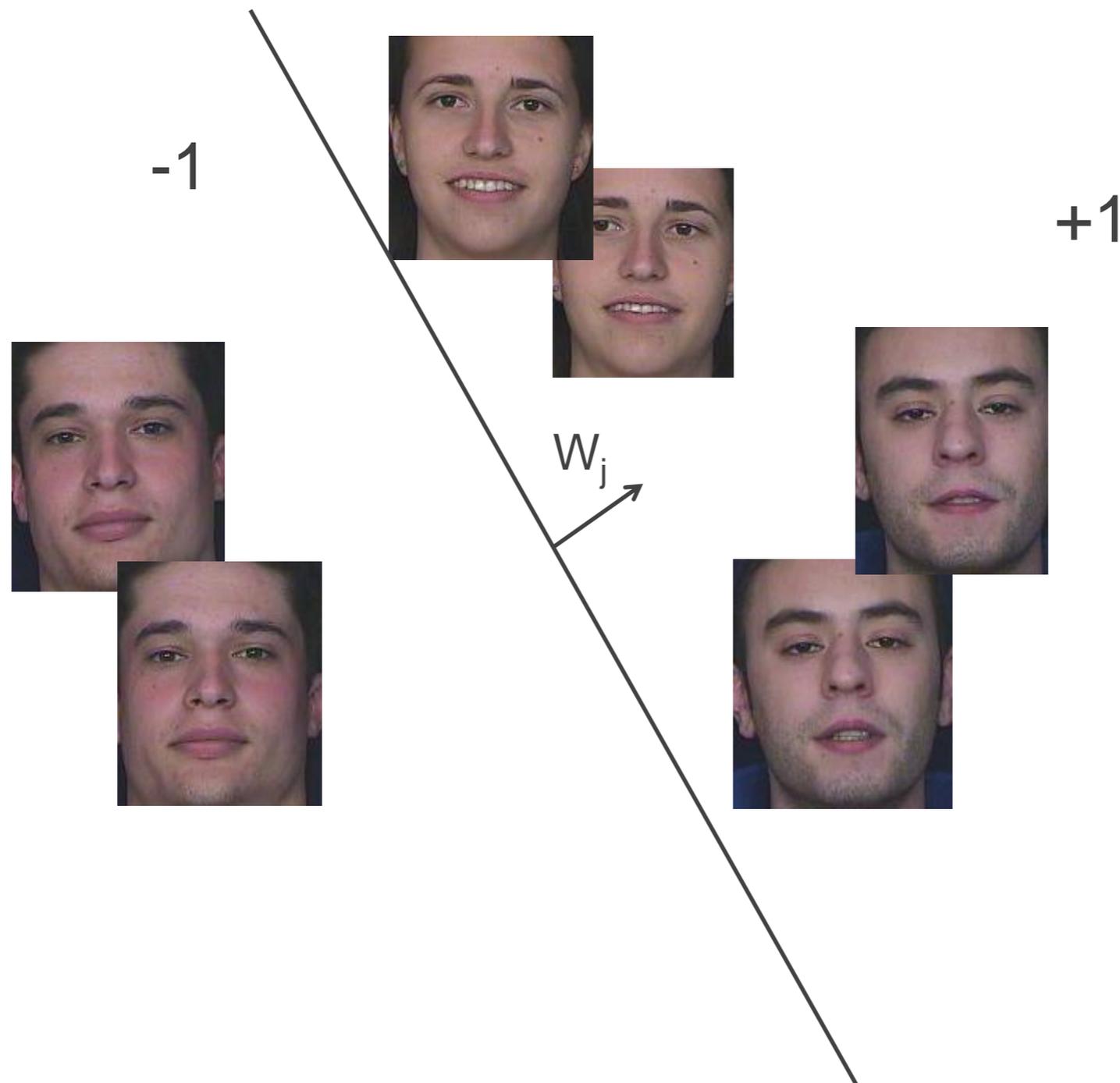
$$h_i(x) = 1$$

$$h_i(x') = 1$$

Embedding in d-dimensional space



Embedding in d-dimensional space



Binarization Alg.

- ❖ Requirements from the binary representation:
 1. Consistency and discrimination
 2. No correlations between the bits
 3. High min-entropy
- ❖ We find a discriminative projection space W by generalizing an algorithm from [WKC10] (for solving ANN problem)
- ❖ For: $X = [x_1, x_2, \dots, x_n]$
 - $(x_i, x_j) \in C$ if the pair belongs to the same user
 - $(x_i, x_j) \in T$ otherwise
- ❖ The aim is to find hyperplanes $[w_1, w_2, \dots, w_K]$, s.t. for: $h_k(x) = \mathbf{sgn}(w_k^t x)$
 - $h_k(x_i) = h_k(x_j)$ if $(x_i, x_j) \in C$
 - $h_k(x_i) \neq h_k(x_j)$ otherwise

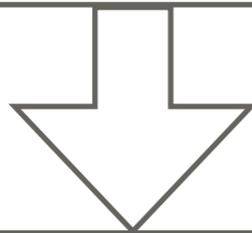
Removing Dependencies between Bits

Dimension Reduction and Concatenation
of Feature Vectors

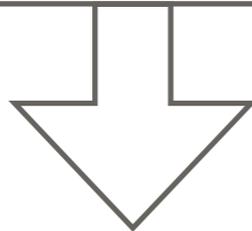


Removing Dependencies between Bits

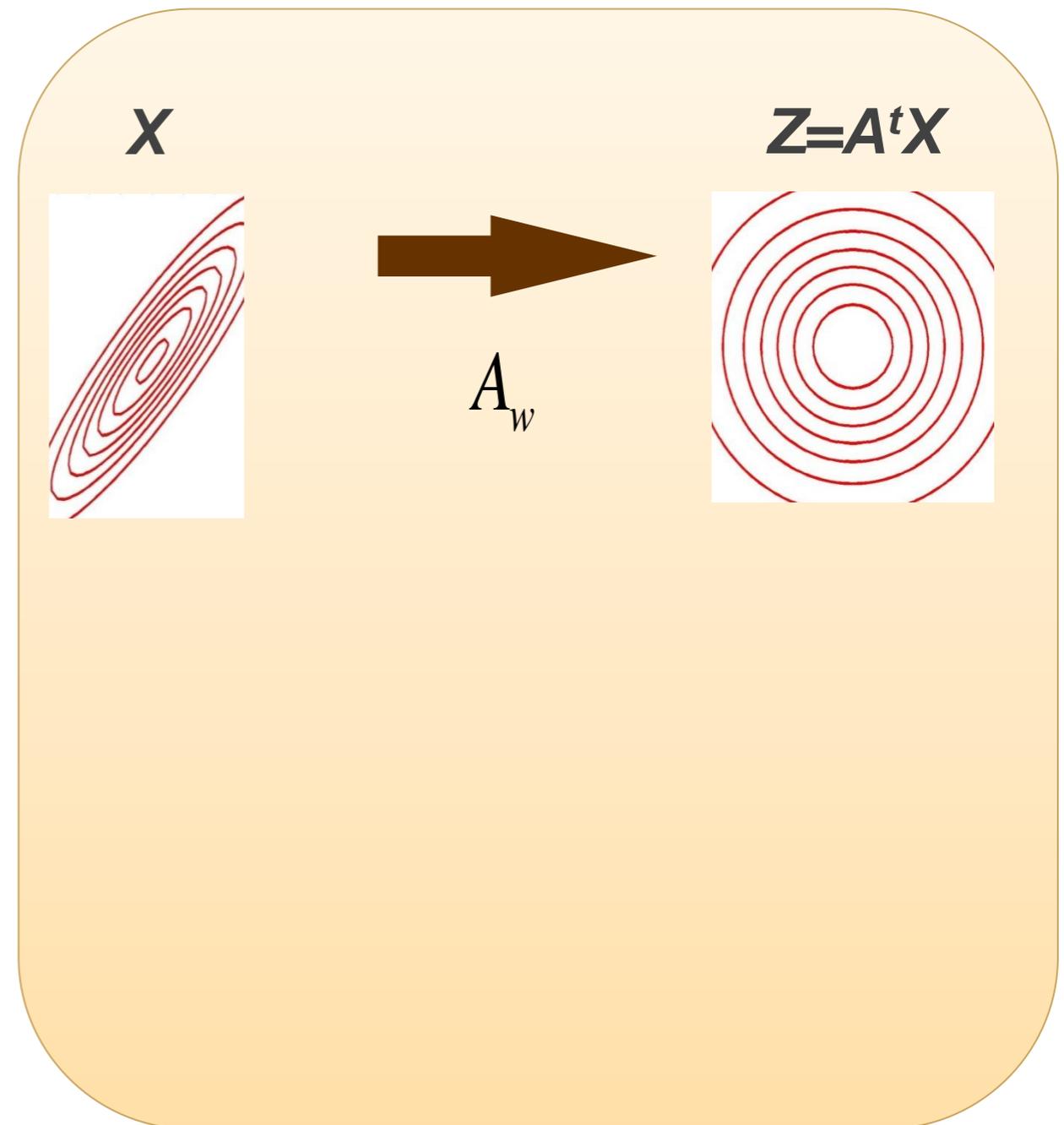
Dimension Reduction and Concatenation
of Feature Vectors



Removing Correlations
Between the Features

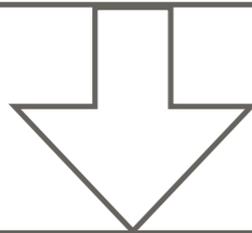


Rescaling for the $[0,1]$
Interval

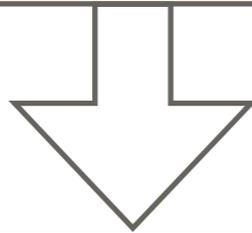


Removing Dependencies between Bits

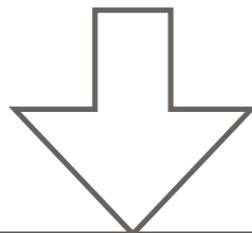
Dimension Reduction and Concatenation
of Feature Vectors



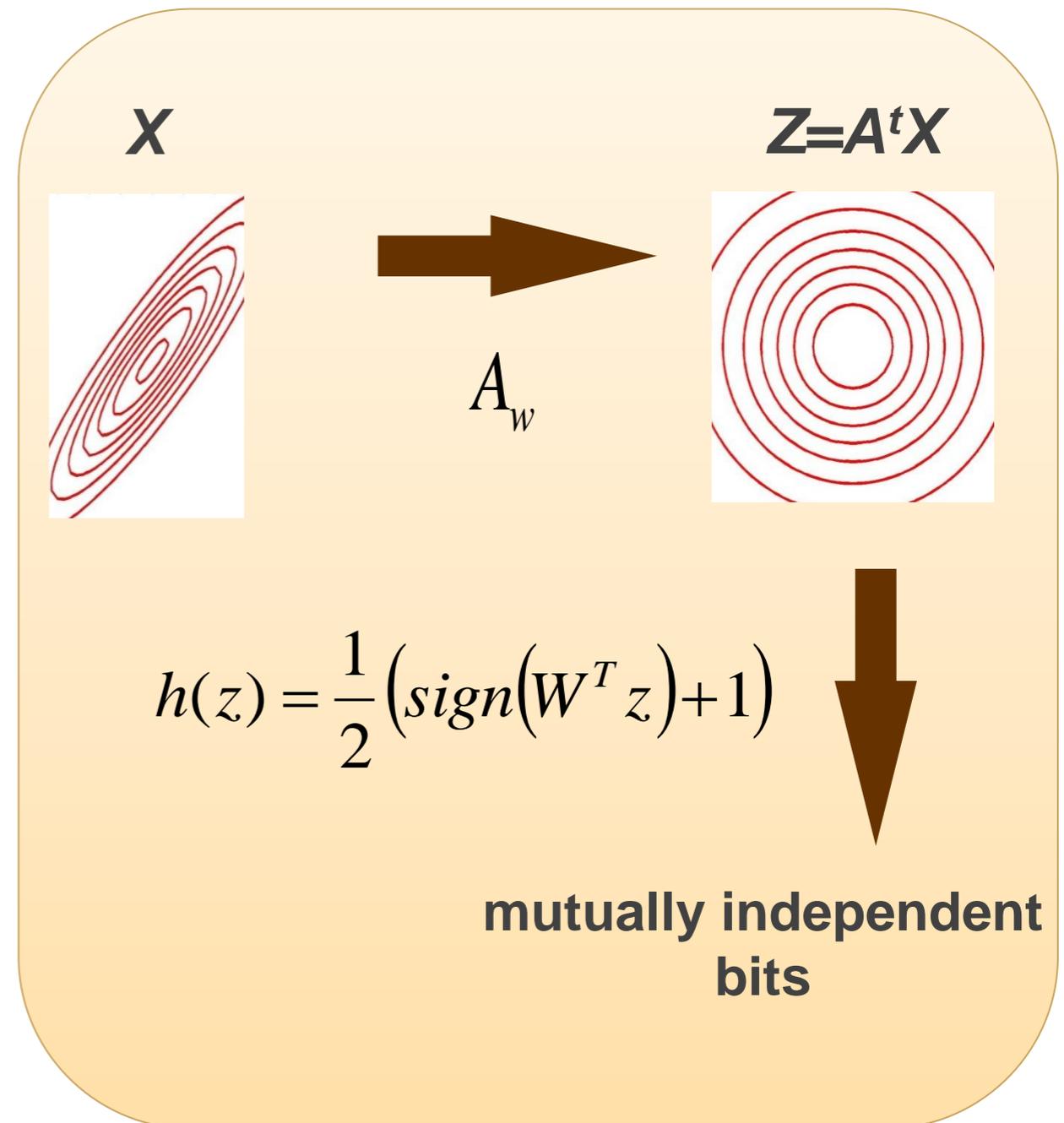
Removing Correlations
Between the Features



Rescaling for the [0,1]
Interval

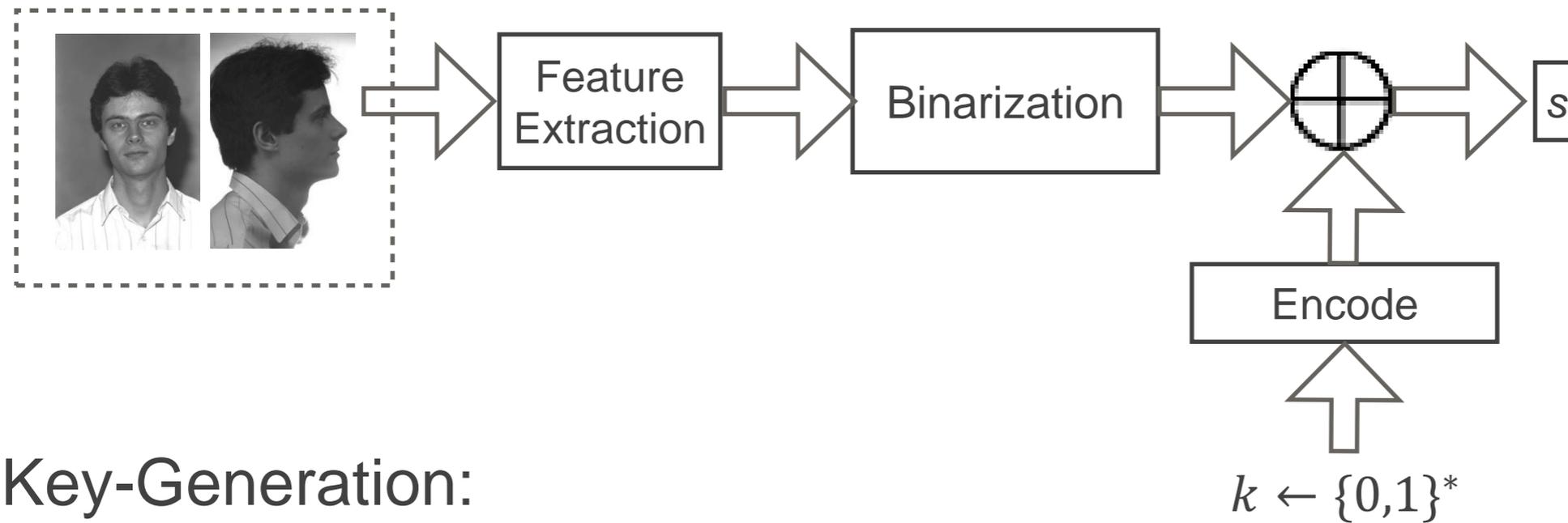


Projection onto orthogonal hypereplanes W

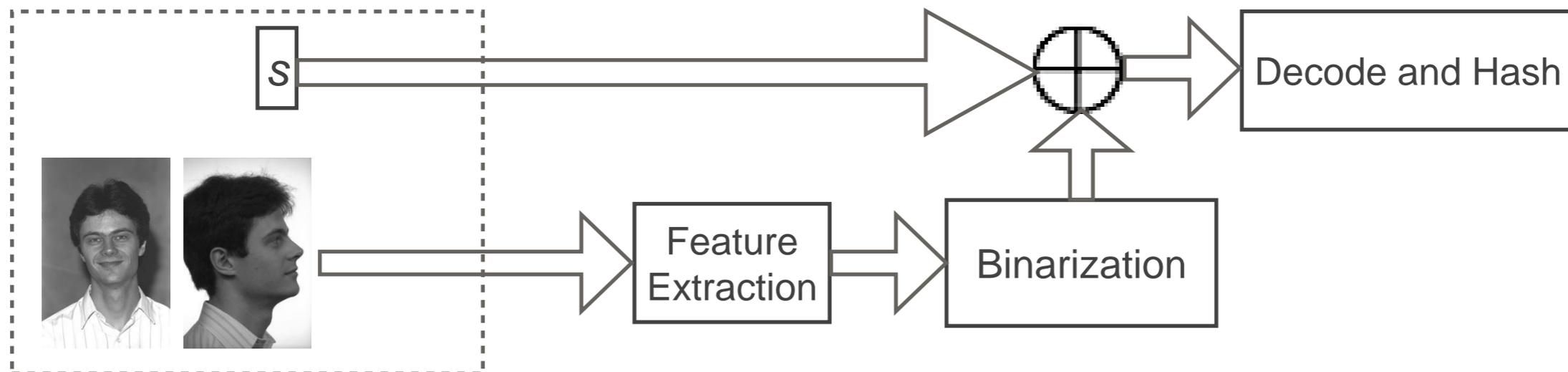


Full System

❖ Enrollment:



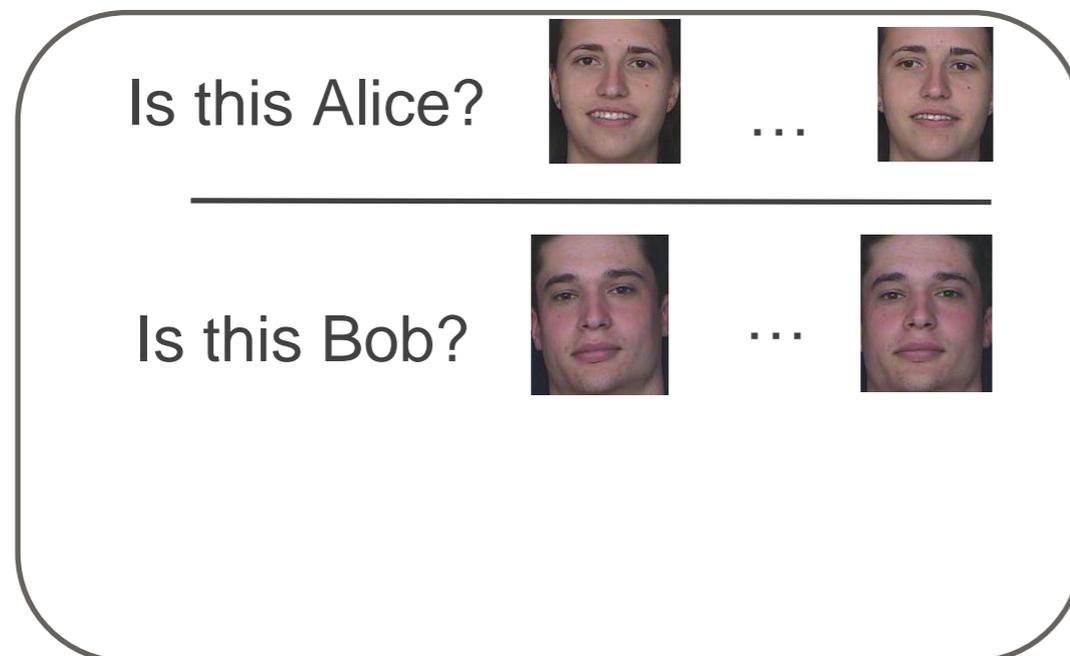
❖ Key-Generation:



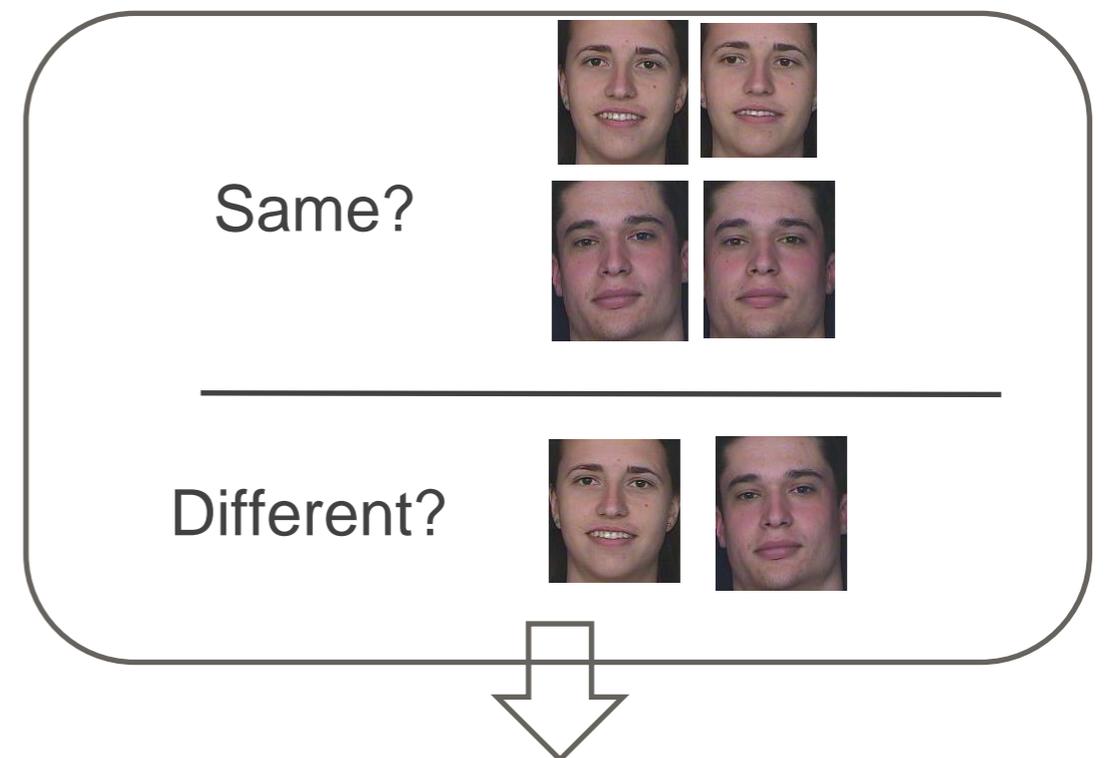
Transfer Learning of the Embedding

- Learning W is done only **once** using subjects **different from the users of the key derivation system.**
- How is it done?

Instead of learning



We learn



A more generic question that can be learnt for population.

Experiments

Constructing the Embedding

- Performed only once
- Subjects are different than those in testing

View	Number of Subjects	Images Per Subject	Number of Hyperplanes
Frontal	949	3-4	800
Profile	1117	1-8	800

Experiments

Evaluation

❖ Data:

- 2 frontal images and 2 profile images of 100 different subjects (not in the training set) were used

❖ Recognition tests:

- 5 round cross validation framework was followed to measure TPR-vs-FPR while increasing the threshold (ROC-curves)

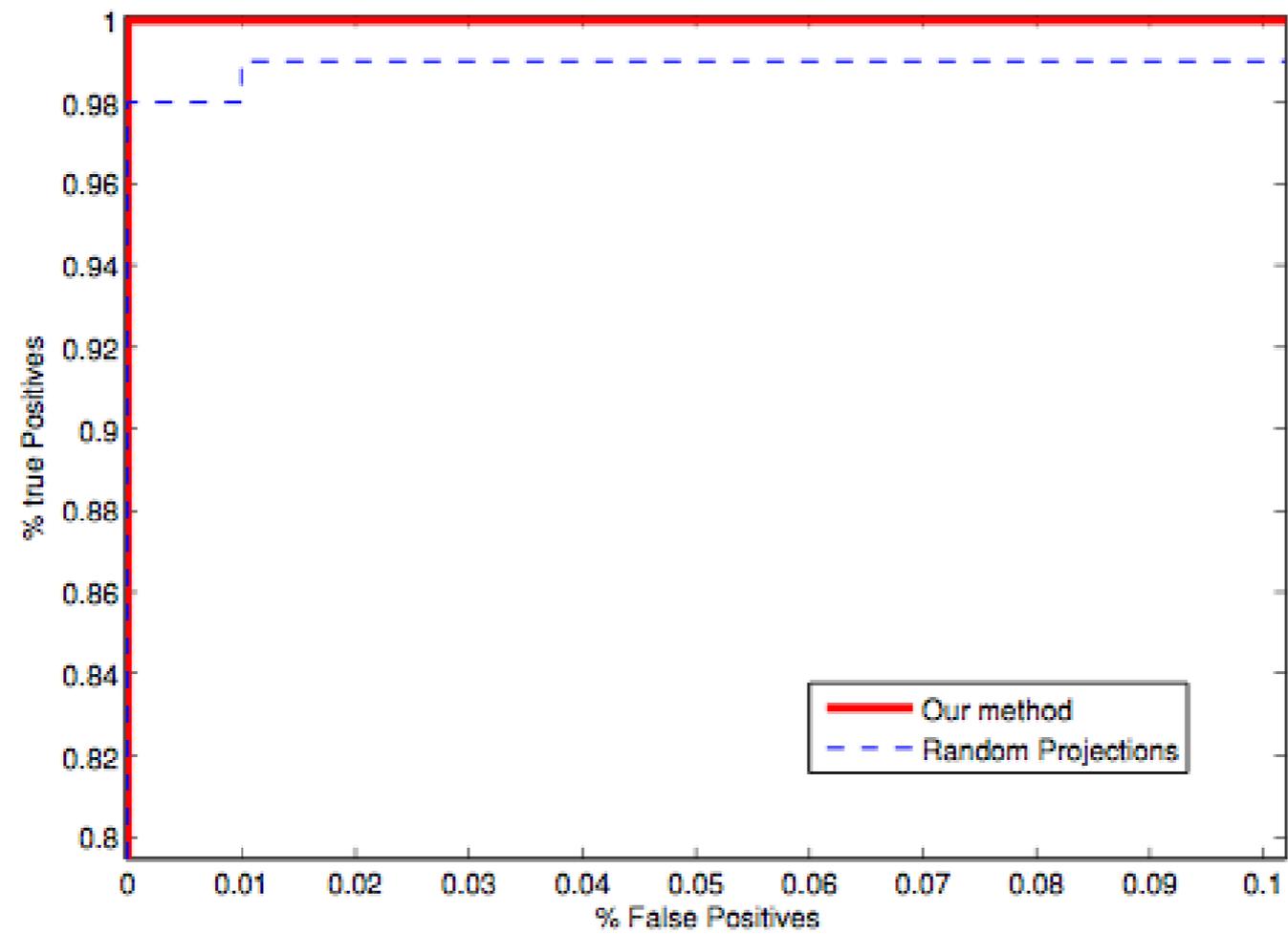
❖ Key generation tests:

- 100 genuine authentication attempts, and 99*100 impostor authentication attempts

Results

Recognition

ROC curves



Results

Key Generation

- ❖ There is a trade-off between the amount of errors that the error-correction code can handle and the length of the produced key
- ❖ The Hamming-bound gives the following relation:

$$k \leq \log_2 \left(\frac{2^n}{\sum_{i=0}^t \binom{n}{i}} \right)$$

- n : the code length ($=1600$ in our case)

- t : the maximal number of corrected errors

- k : the length of the encoded message (produced key, in our case)

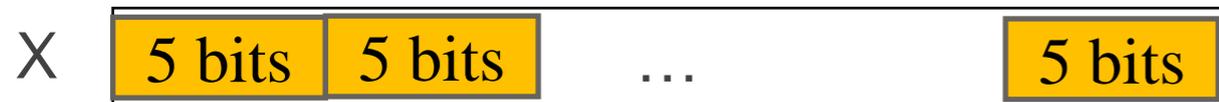
Results

Key Generation

For $FAR=0$

t	$k \geq$	FRR our method	FRR Random Projection
595	80	0.30	0.32
609	70	0.16	0.23
624	60	0.12	0.19

Error Correction Code

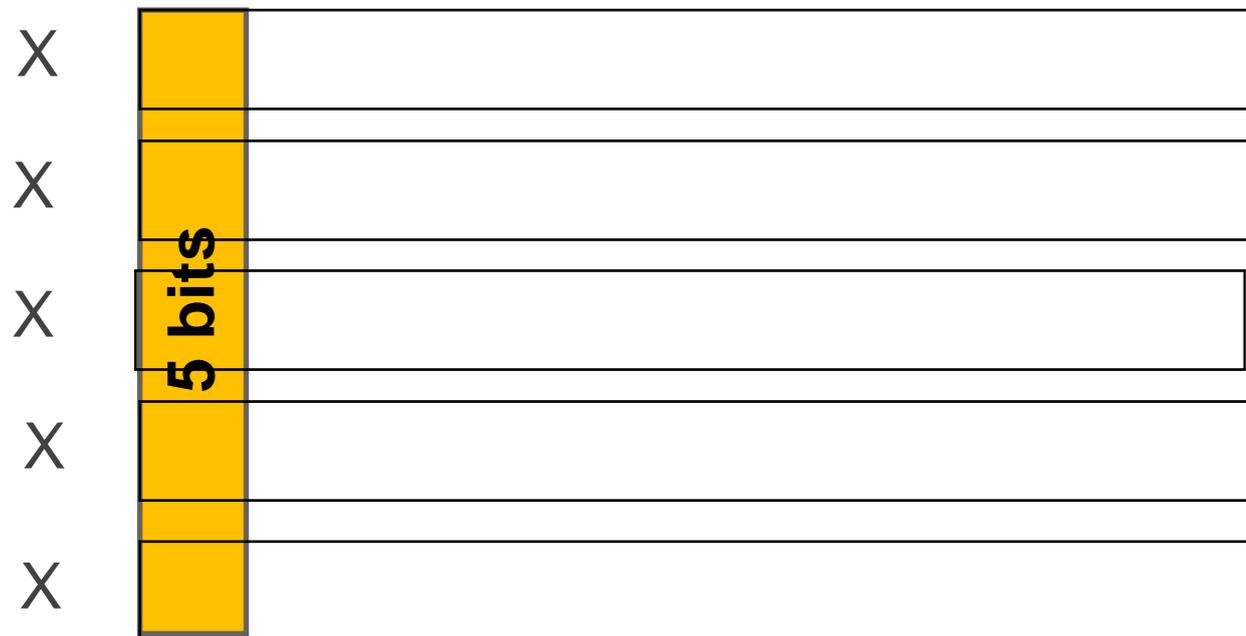


RS,GF(2⁵): 15,GF(2⁵) \Rightarrow 31,GF(2⁵)

Probability of error in bit 0.3 \Rightarrow Probability of error in symbol $1-0.7^5 \approx 0.83$



Possible Solution

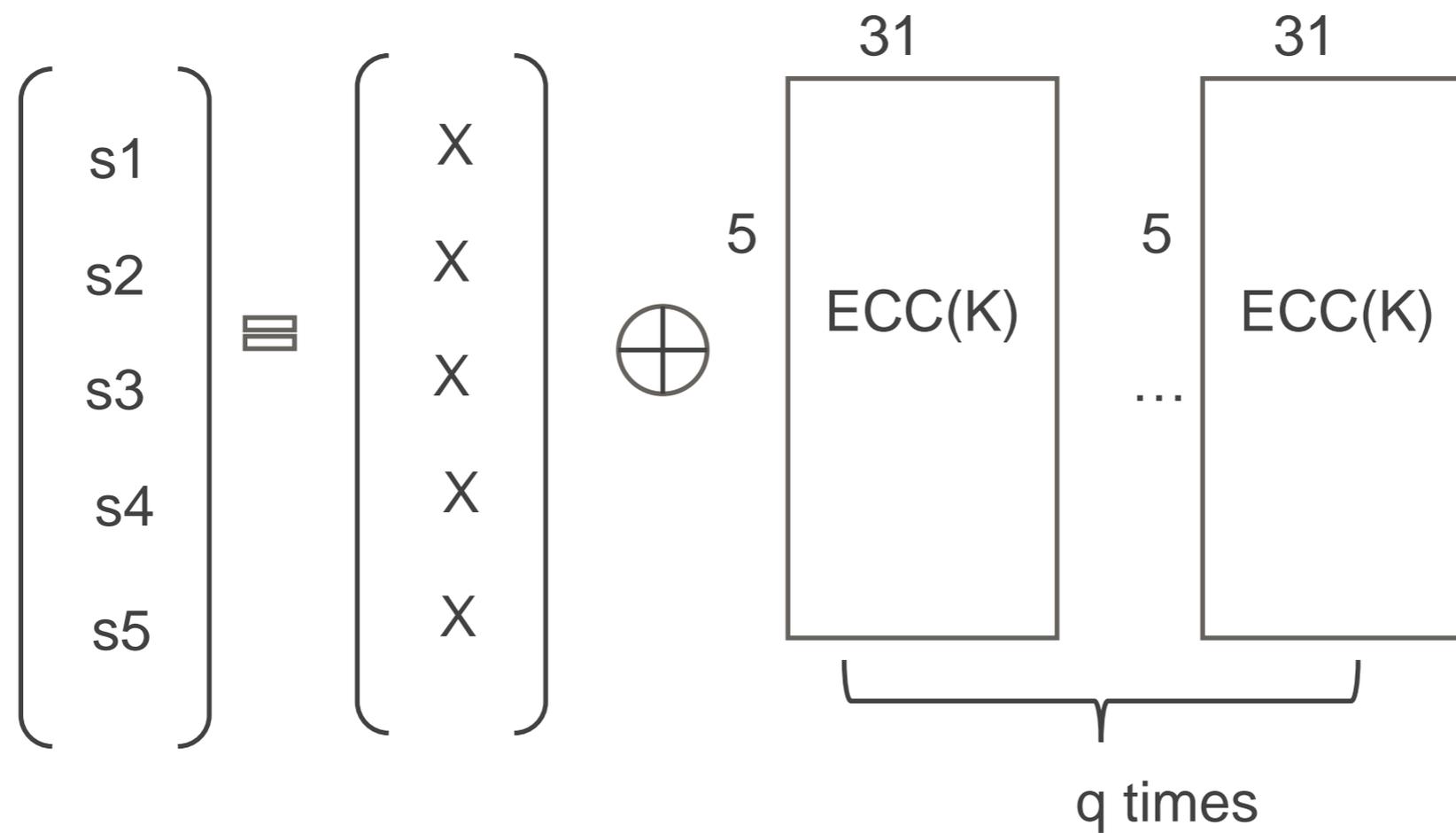


RS,GF(2⁵): 15,GF(2⁵) \Rightarrow 31,GF(2⁵)

Probability of error in bit 0.3 \equiv Probability of error in symbol 0.3

Possible Solution

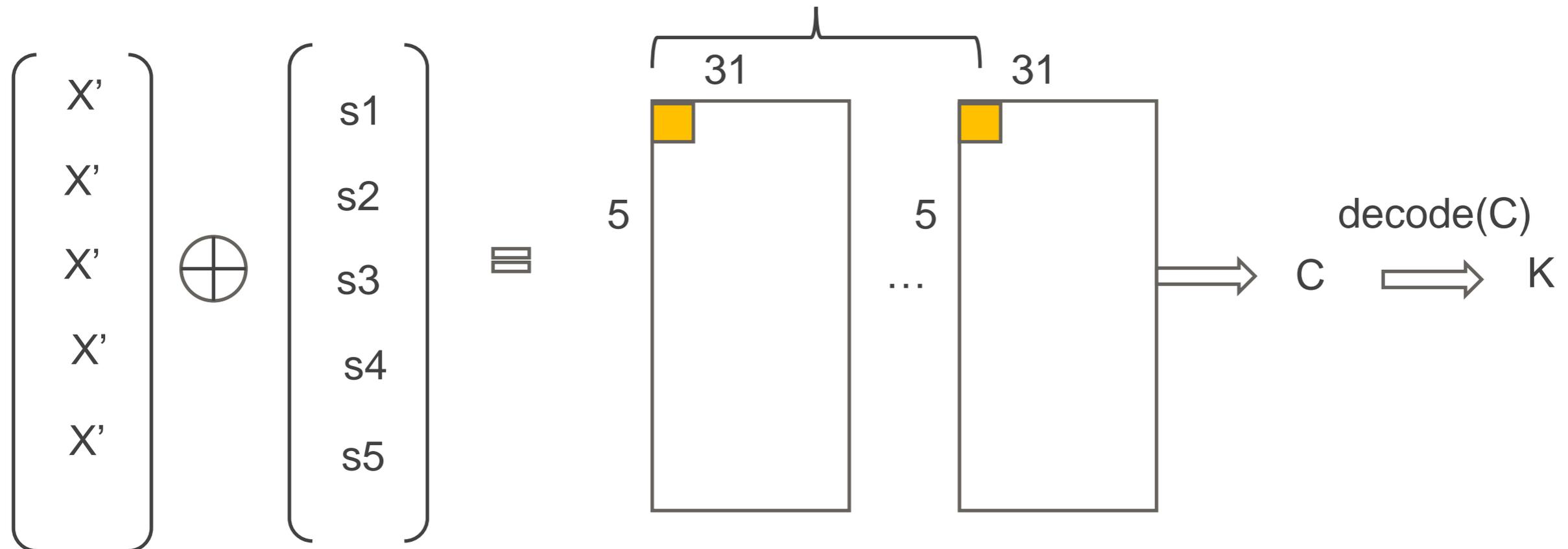
Encoding:



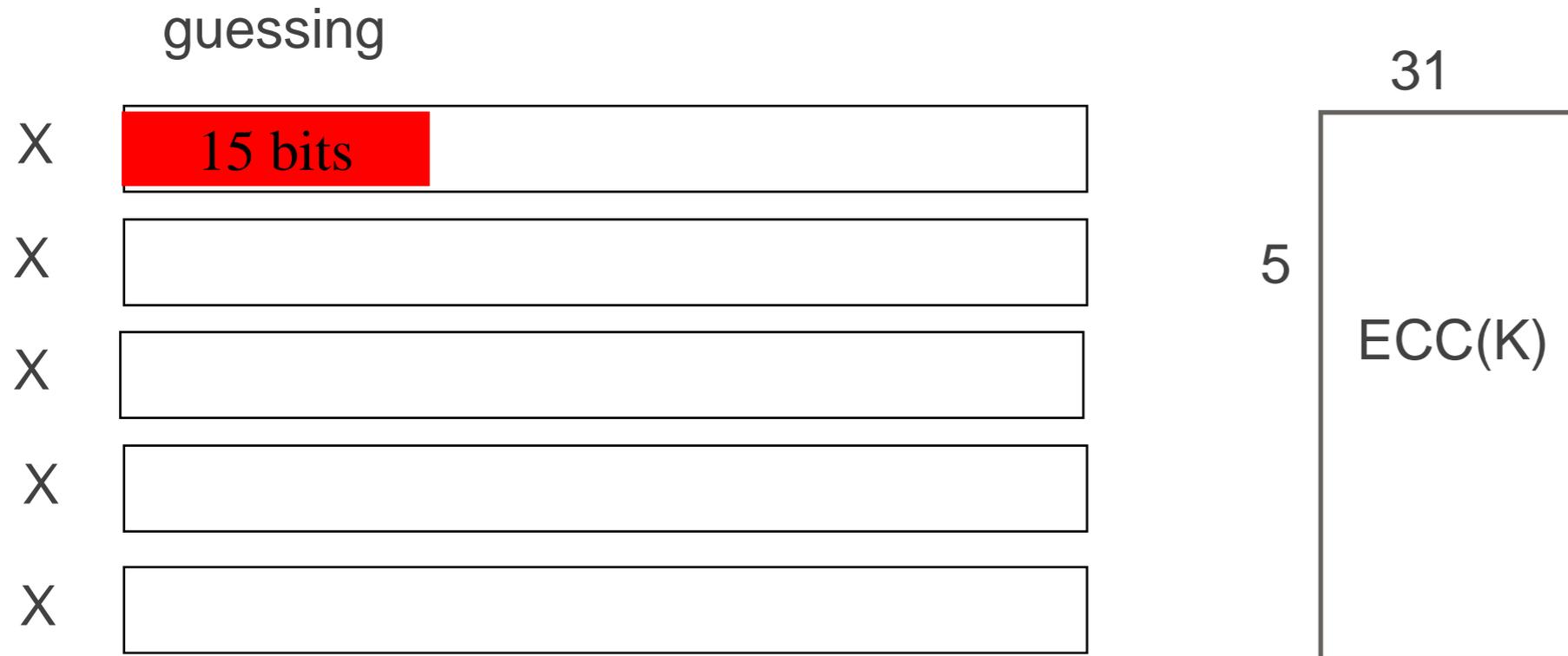
Possible Solution

Decoding:

the value of the bit = majority over q values

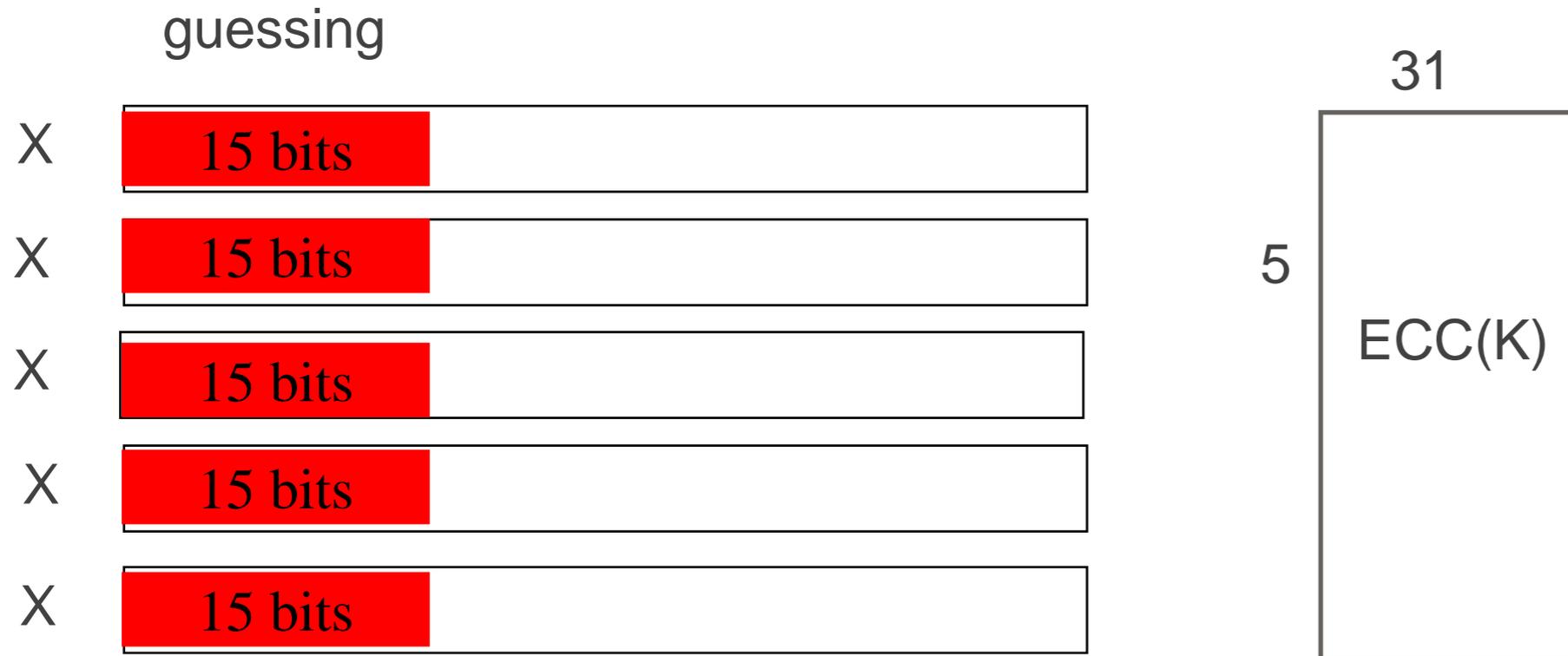


Problem in Security



RS,GF(2⁵): 15,GF(2⁵) \Rightarrow 31,GF(2⁵)

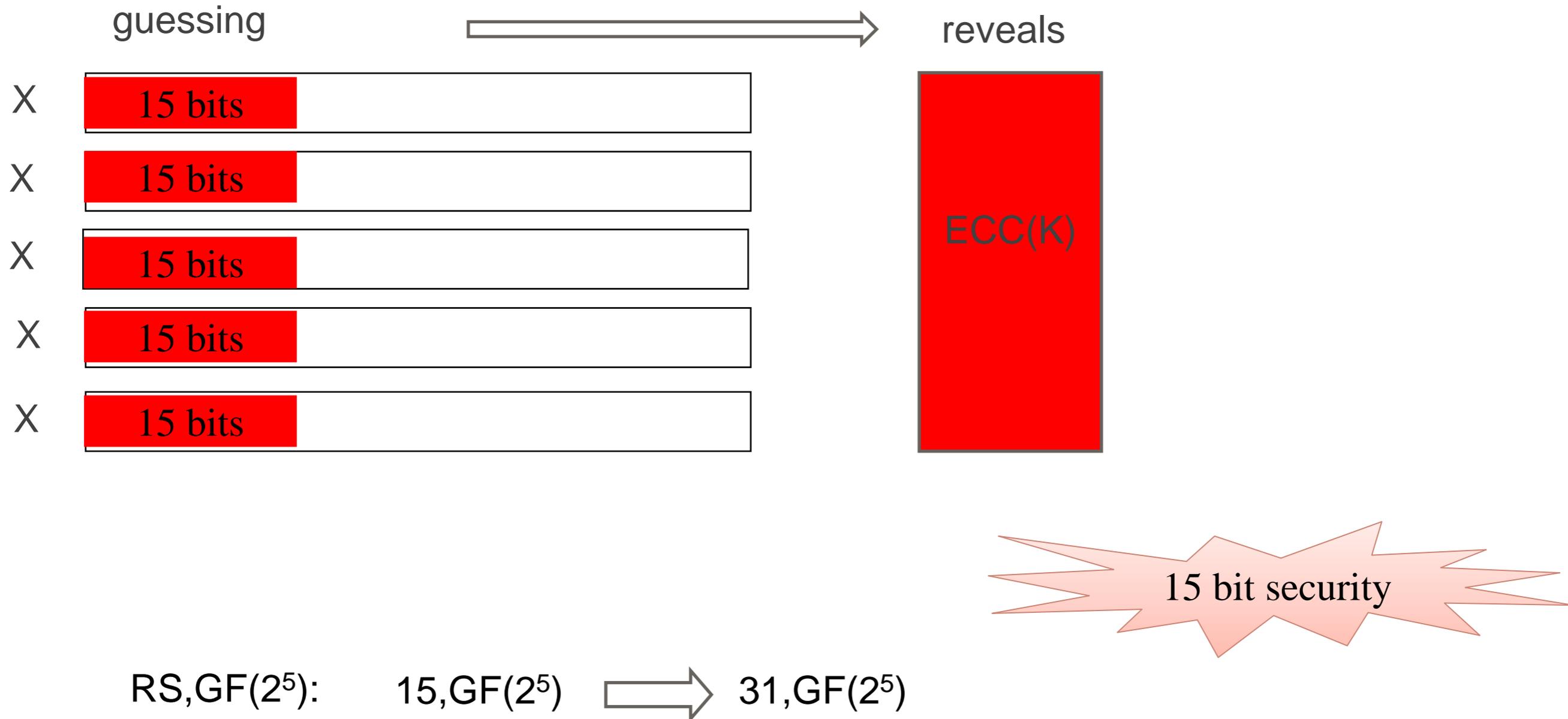
Problem in Security



RS,GF(2⁵): 15,GF(2⁵) \Rightarrow 31,GF(2⁵)

Probability of error in bit 0.3 \equiv Probability of error in symbol 0.3

Problem in Security

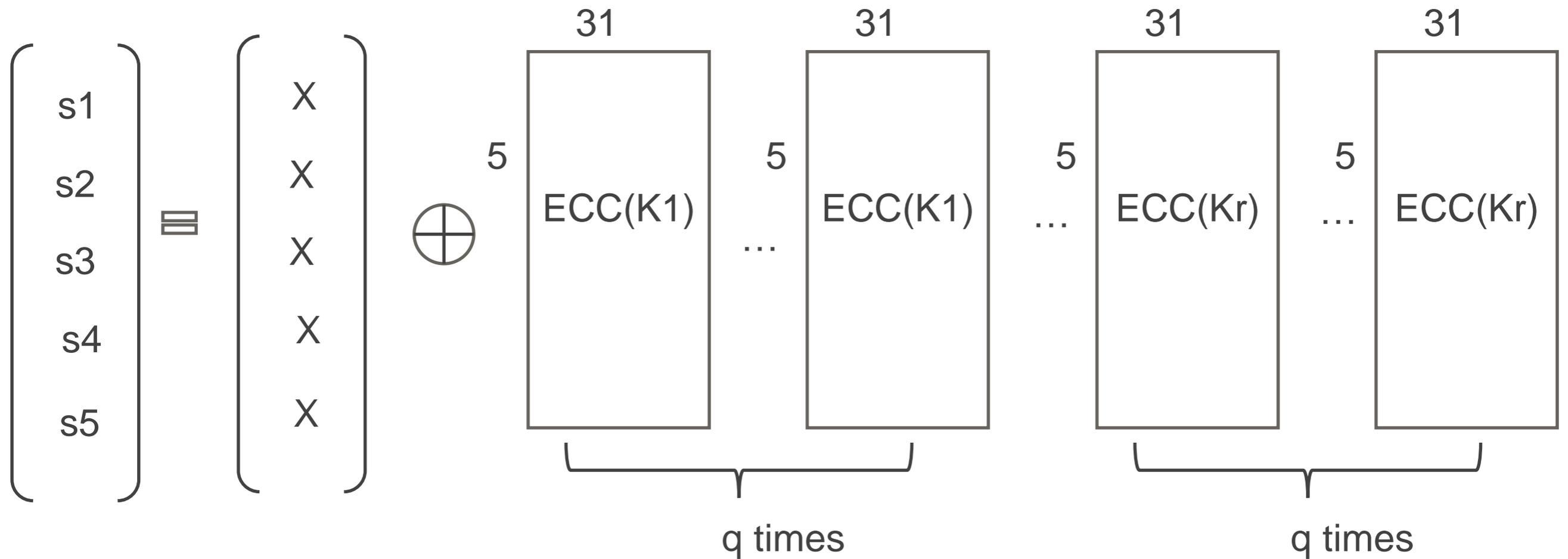


Secret Sharing Solution

RS, GF(2⁵): 15, GF(2⁵) \Rightarrow 31, GF(2⁵)

15r bit security

$$K = K_1 \oplus K_2 \oplus \dots \oplus K_r$$



Results for $r=3$

Key Length	75 bits
Security	45 bits
FRR	0.085
FAR	5.6022e-04
Encoding time (Matlab implementation)	0.07 sec
Decoding time (Matlab implementation)	0.05 sec

Security Analysis

1. Consistency: $FRR = 0.085$ (for 75-bit key)
2. Discrimination: FAR very low
3. REQ-WBP: follows from REQ-SBP
4. REQ-SBP: this property is accomplished if the representation is uniformly distributed, as shown in [JW99]

Security Analysis

Uniformity of the Representation

No correlation between the bits + high min-entropy \Rightarrow uniform distribution

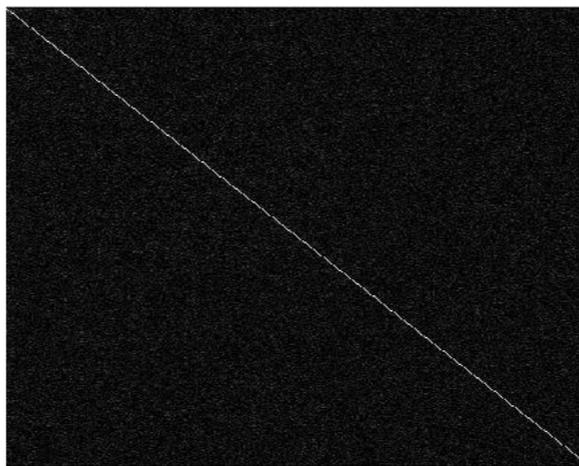
- ❖ No correlation between the bits - way :1
 - High degrees-of-freedom ($\gamma = \frac{p(1-p)}{\sigma^2}$): 1571.72
 - p : average relative distance between two representation of different persons
 - σ the standard deviation

Security Analysis

Uniformity of the Representation

No correlation between the bits + high min-entropy \Rightarrow uniform distribution

- ❖ No correlation between the bits – way 2:
- ❖ The representation has a diagonal covariance matrix:



- ❖ High min-entropy: 1562.02
(maximal bias from 0.5 is 0.0757, average distance from 0.5 is 0.0066)

Security Analysis

1. Consistency: $FRR = 0.16$ (for 70-bit key)
2. Discrimination: $FAR = 0$
3. REQ-WBP: follows from REQ-SBP
4. REQ-SBP: this property is accomplished if the representation is uniformly distributed, as shown in [JW99]
5. REQ-KR: next

Security Analysis

REQ-KR

- ❖ Show that $H_\infty(k|s)$ is high

$$k = \text{decode}(x \oplus s)$$

- ❖ $x \sim U \rightarrow$ all possible results of $\text{decode}(x \oplus s)$ have an almost equal probability, regardless of s 's value
- ❖ Thus, $H_\infty(k|s) = H_\infty(\text{decode}(x \oplus s)|s) = H_\infty(\text{decode}(x \oplus s))$ is high

Conclusions

- ❖ We showed a system for Key-Derivation that achieves:
 1. Consistency and discriminability
 2. High min-entropy representation
 3. Provable security
 4. Provable privacy
 5. Fast face-authentication

What this is Good for?

- ❖ Key derivation schemes – your face is your key
- ❖ Can be easily transformed into a login mechanism
- ❖ Can be used in biometric databases (identify double acquisition without hurting honest users' privacy)

Future Work

1. Generating longer keys - by decreasing the distance within the same class/subject
2. Adding invariance to changes in viewing conditions and intra-personal changes
3. Improving the error-correction

Thank You!