Shannon Information theory, coding and biometrics

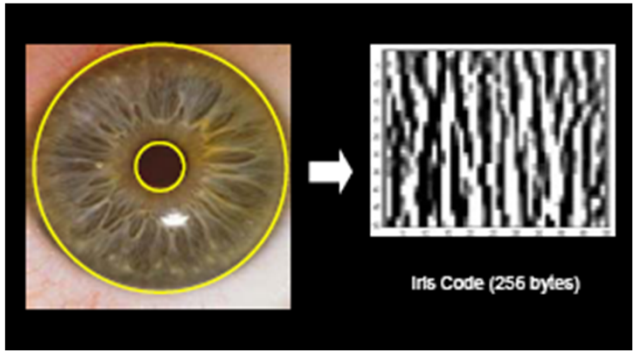
Han Vinck

June 2013

We consider

- The password problem using biometrics
- Shannon‘s view on security
- Connection to Biometrics

Goal: use biometrical features as passwords



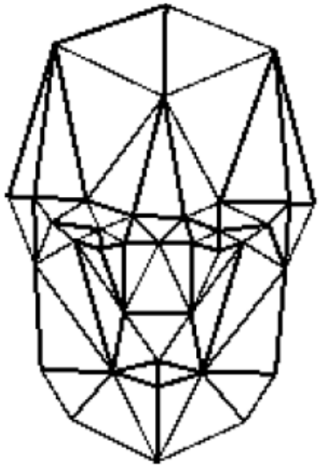


Illustration of the password problem

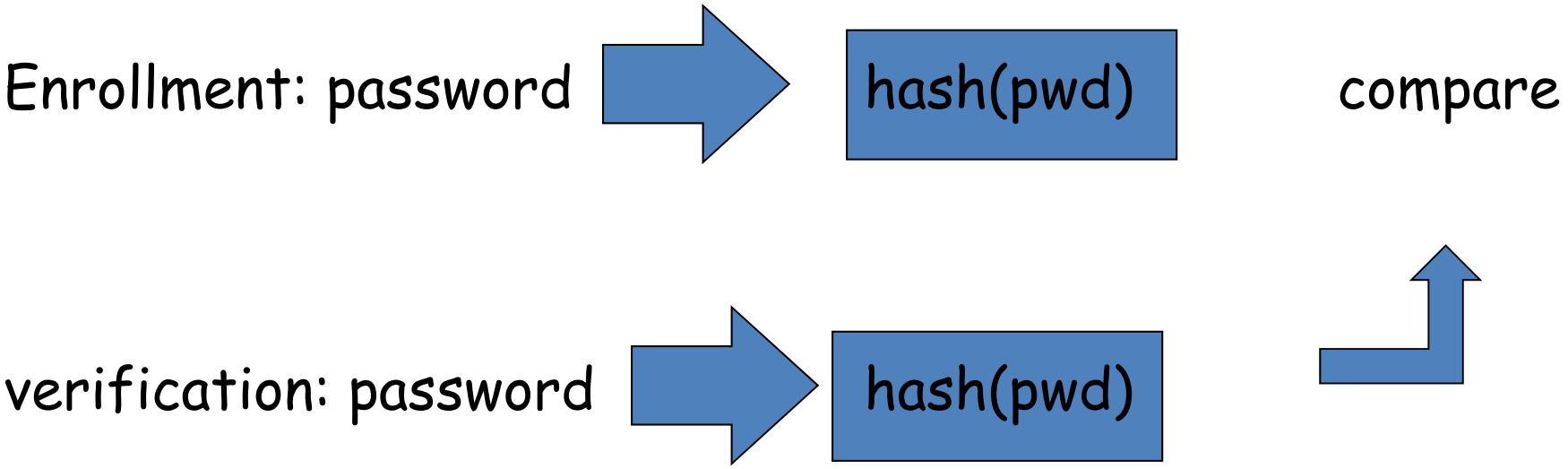
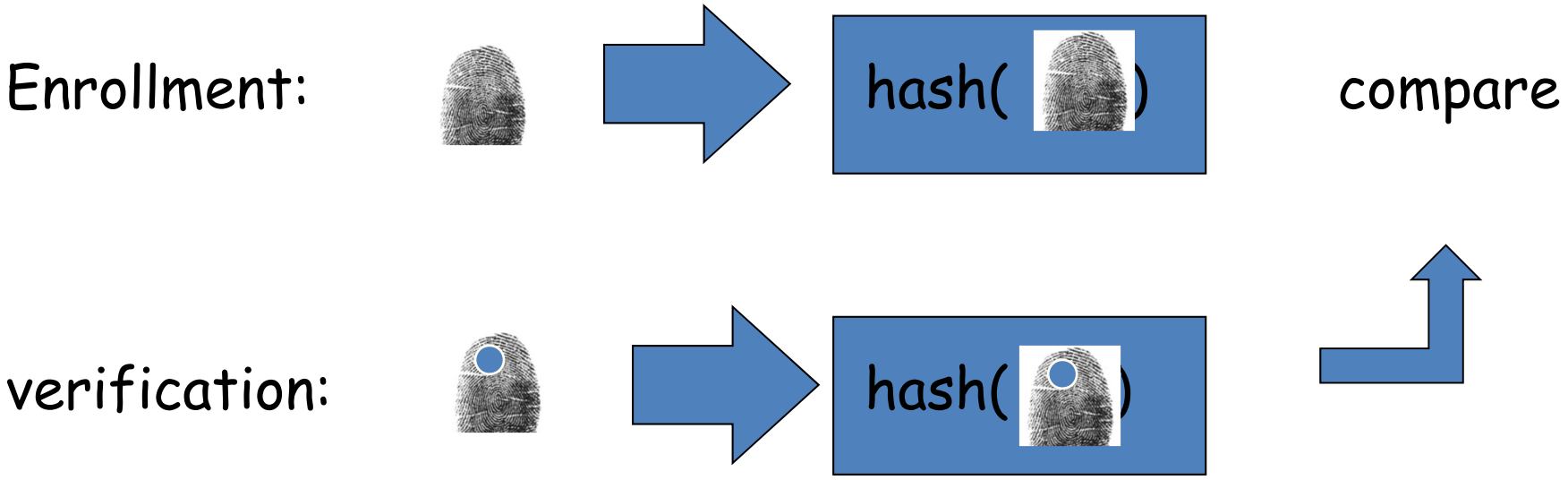


Illustration of the problem



hash functions of biometrics can not be used as passwords

for a vector c and a noisy version $c' = c \oplus \text{noise}$

hash property: $\text{hash}(c' \approx c) \neq \text{hash}(c)$

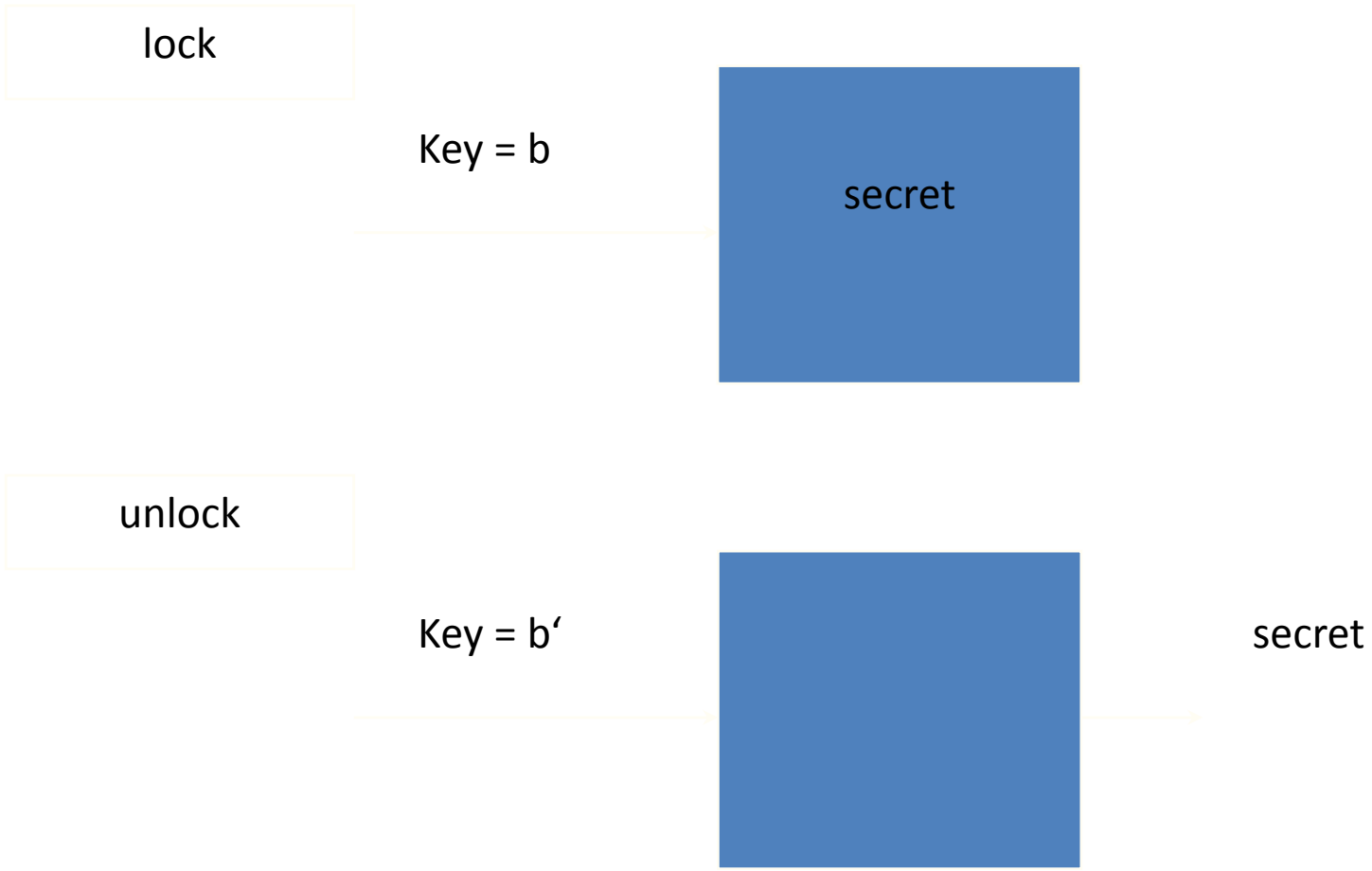
single error => n/2 differences

may be we can use Error-correction:

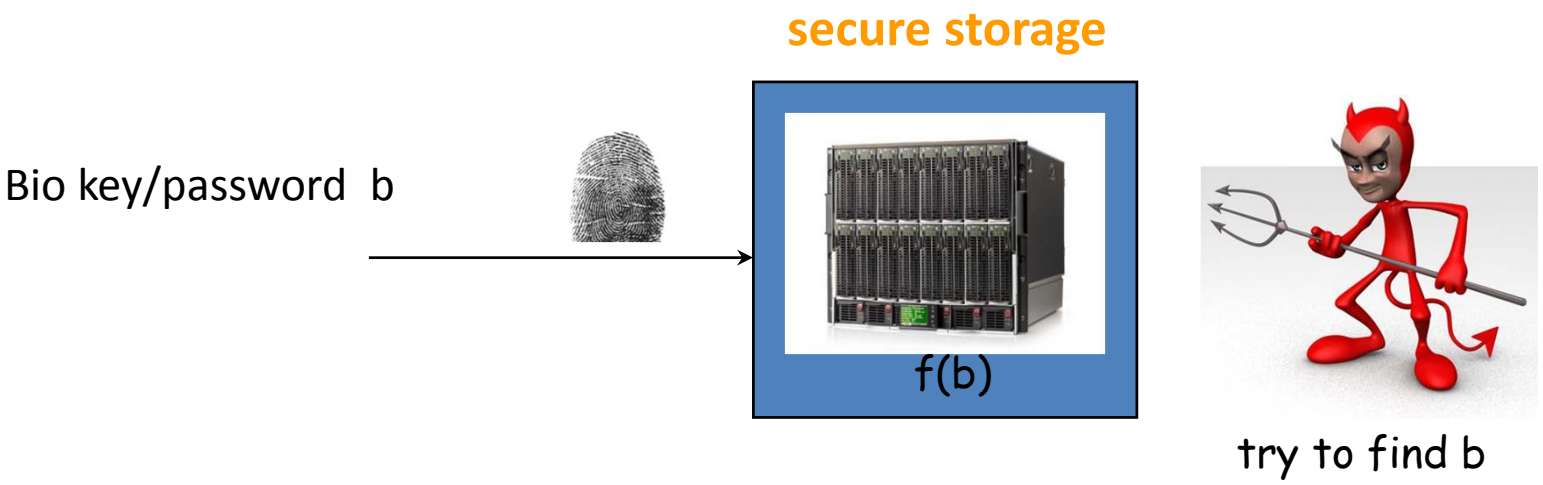
$\text{dec}(c' \approx c) = \text{dec}(c)$

equality for $2t < d_{min}$

This is what we want



Problem: secure storage and biometric authentication



biometrics

- Definition:

  Methodology for recognizing and identifying people based on individual and distinct physiological or behavioral characteristics

biometrics

- Authentication through

  - learned skils:
    - such as recognition of speech,
    - dynamics of signature,
    - keystroke patterns

  - Natural properties such as
    - Fingerprints
    - Iris pattern
    - Retina, hand geometry
    - Facial scan
    - etc.

http://www.youtube.com/watch?v=BufSl0VurHo&feature=related

Hand Geometry

Popular form of biometric

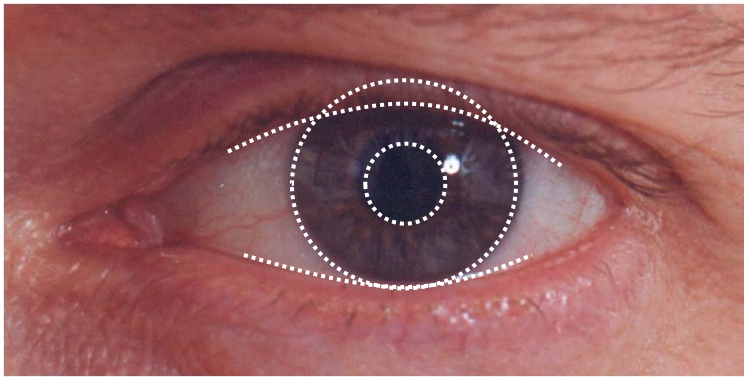
Measures shape of hand
- Width of hand, fingers
- Length of fingers, etc.

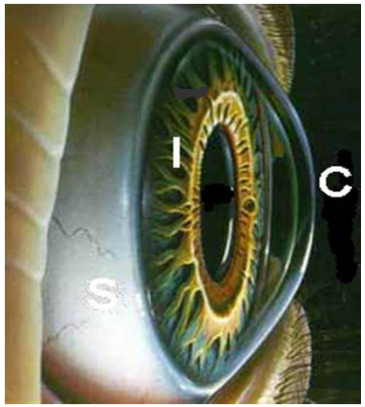
Human hands not unique

Hand geometry sufficient for many situations

Suitable for authentication

Iris Patterns





- Iris pattern development is "chaotic"
- Little or no genetic influence
- Different even for identical twins
- Pattern is stable through lifetime

biometrics

- Why?
  - it is a key connected to a person: are always with you
  - universal
  - easy to collect data for enrollment
  - no memorization of voice, face, eyes, or fingerprints
  - are personal: Cannot be given to somebody else

- Problems?
  - sensors needed without medical risk
  - reference values may be not actual (ageing)
  - failure rate rather high
  - passwords are exact, biometrics only approximately

- system requirements: accuracy, speed, complexity

- user requirements: harmless, accepted, robust to attacks

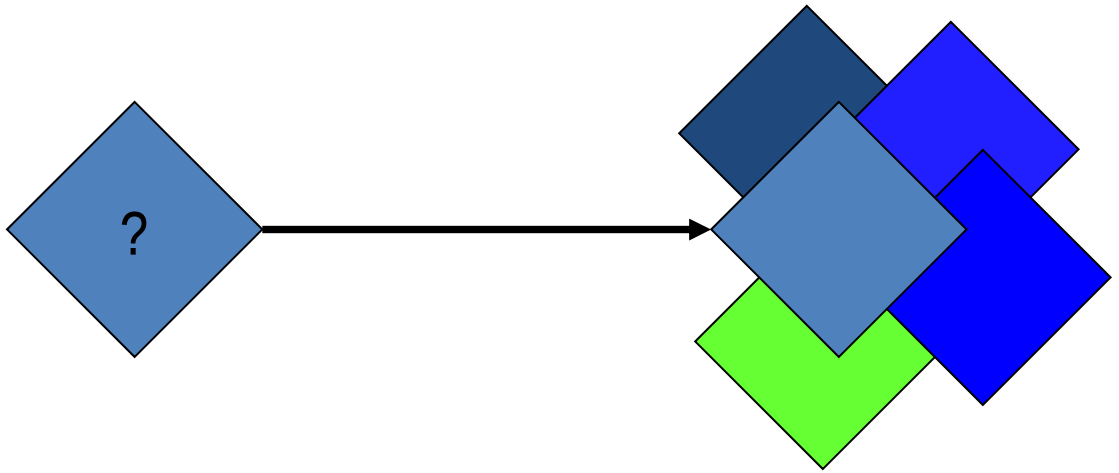
biometrics

- IDENTIFICATION: compare one to many
  - Who goes there?

- AUTHENTICATION: compare one to one
  - Is that really you?

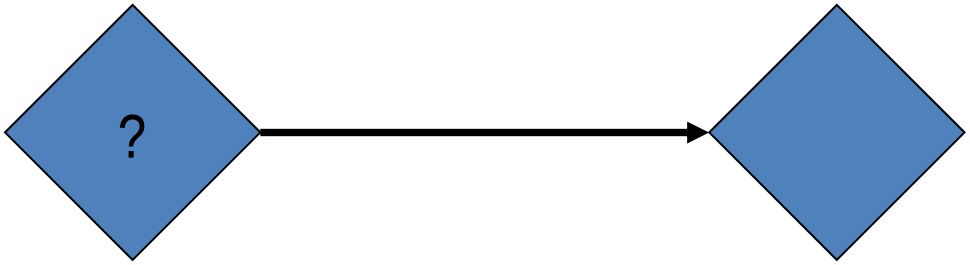
Identification

- Search a sample against a database of templates.
- Typical application: identifying fingerprints

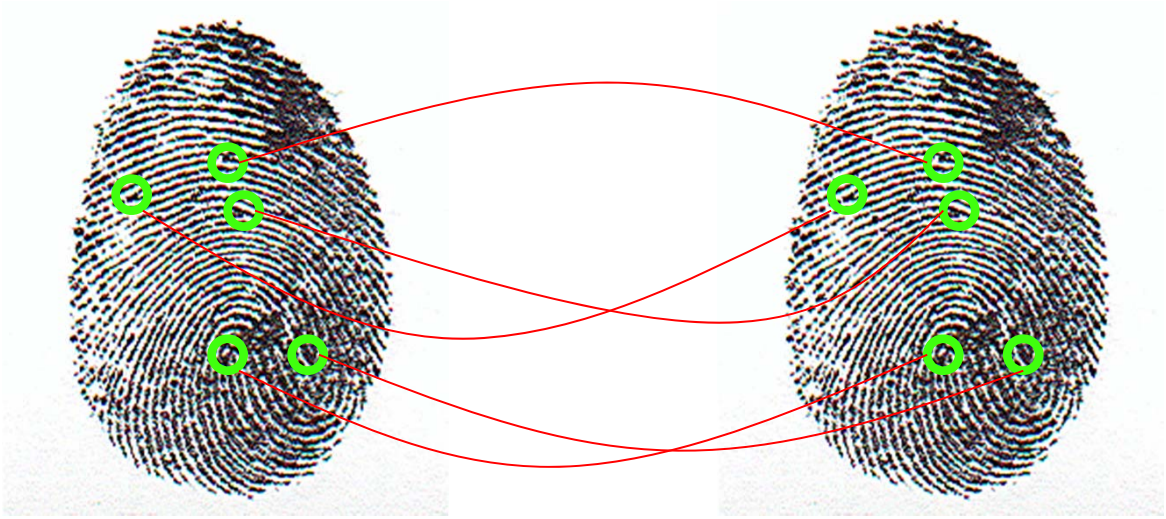


Authentication

- Compare a sample against a single stored template
- Typical application: voice lock



Biometric Fingerprint



- Extracted minutia are compared with user's minutia stored in a database
- Is it a statistical match?

Matching problem

For example: rotation and translation

classification

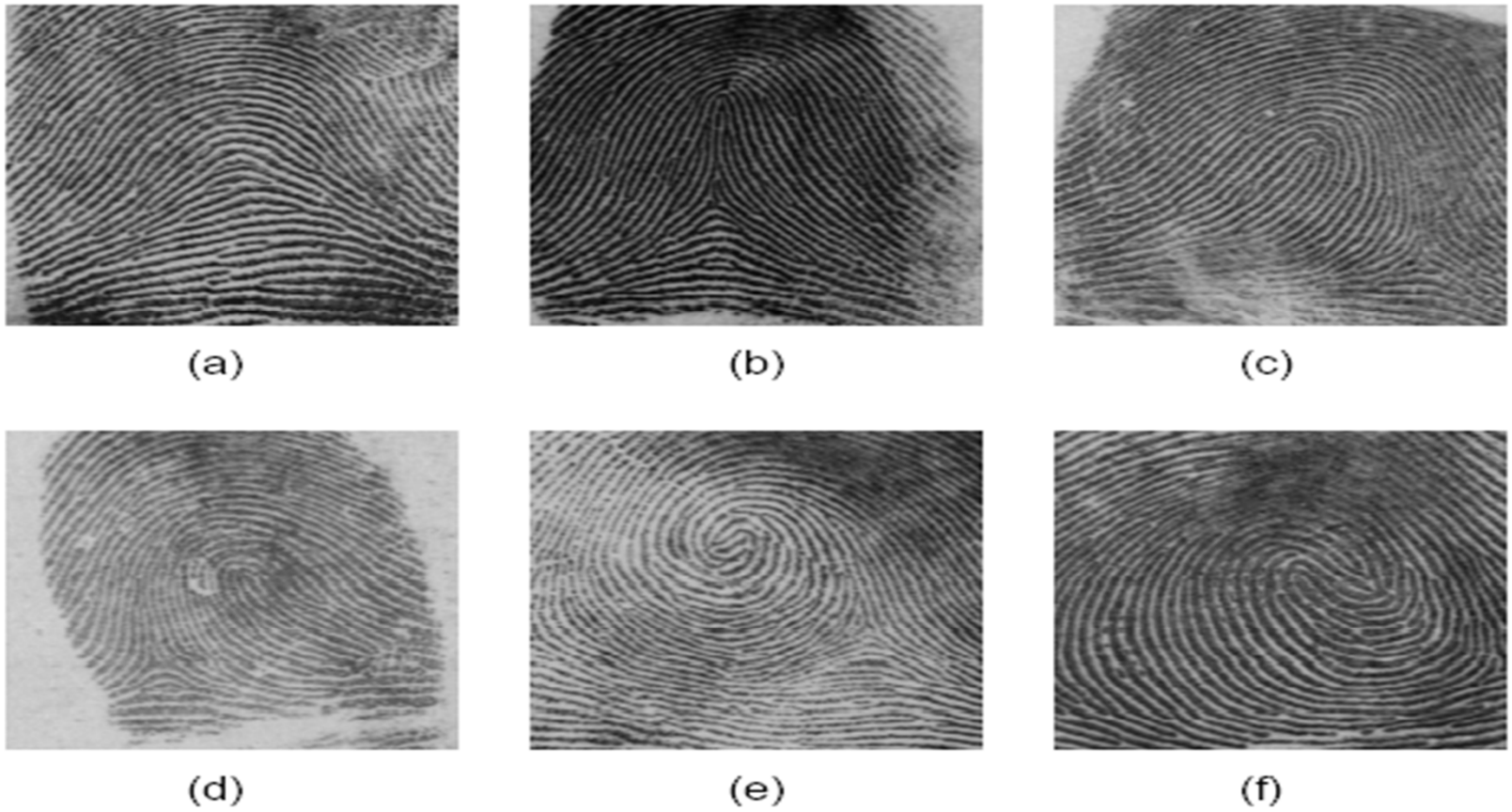
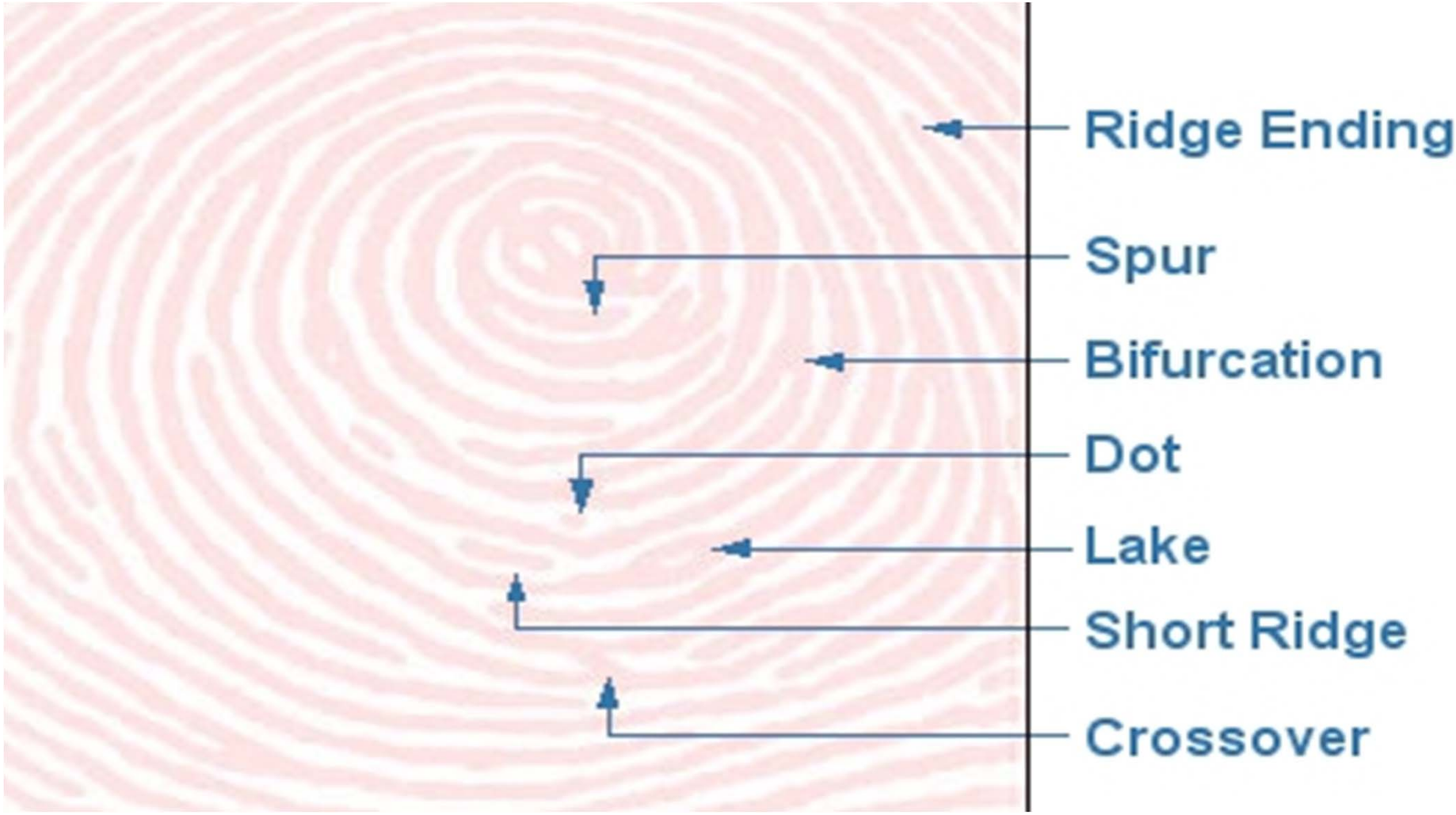


Fig. 3. A coarse-level fingerprint classification into six categories: (a) Arch. (b) Tented arch. (c) Right loop. (d) Left loop. (e) Whorl. (f) Twin loop.

Minutiae (Pavel Margolin)



Minutiae	Example	Minutiae	Example
ridge ending		bridge	
bifurcation		double bifurcation	
dot		trifurcation	
island (short ridge)		opposed bifurcations	
lake (enclosure)		ridge crossing	
hook (spur)		opposed bifurcation/ridge ending	

2 examples of Minutiae

Minutiae can be represented by the location *(x,y)* and the ridge direction

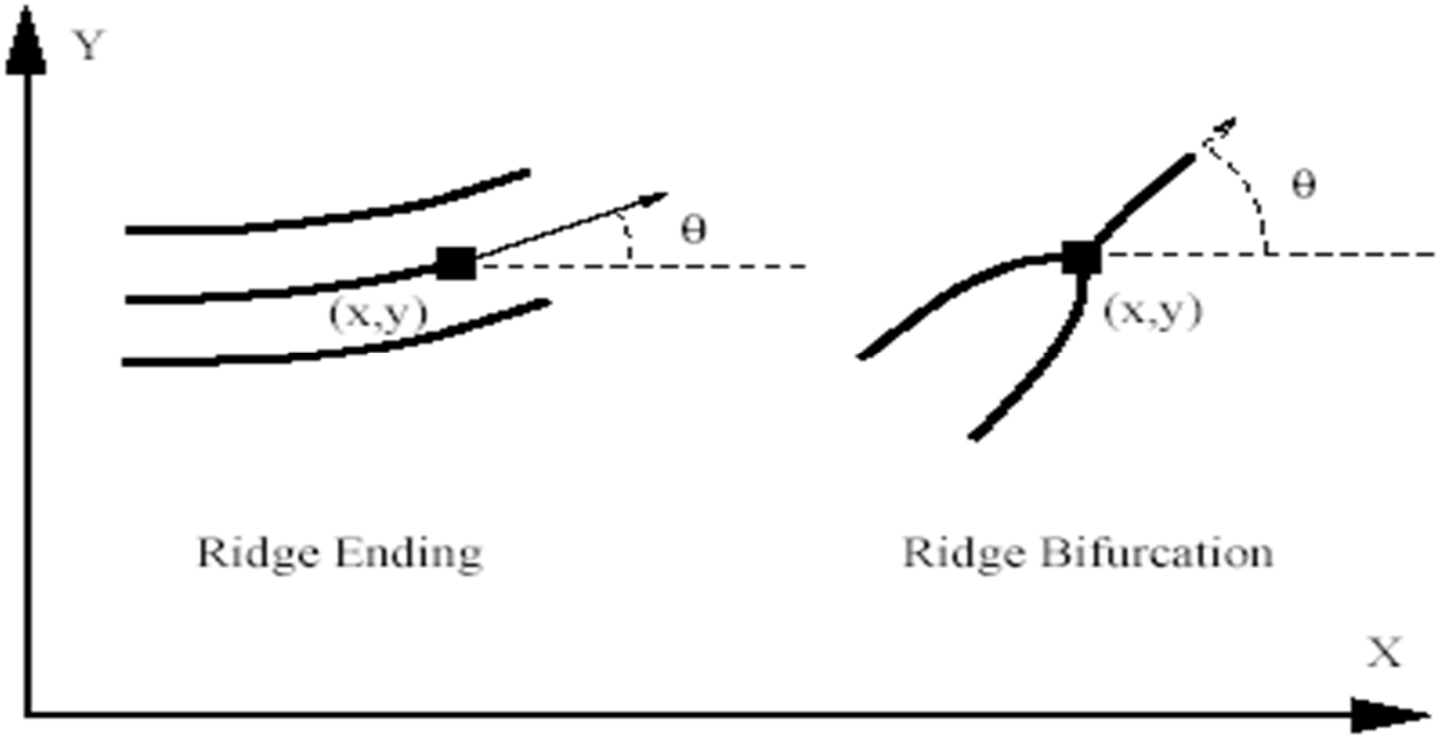


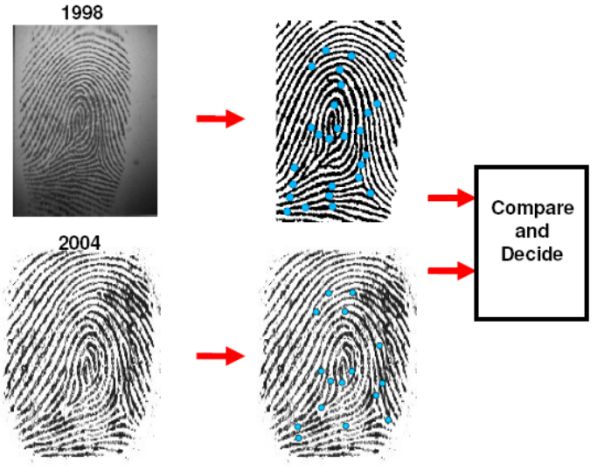
Figure taken from Nandakumar, et al. http://www.cse.msu.edu/~nandakum/FingerprintMatching.ppt

Problem: biometrics do change

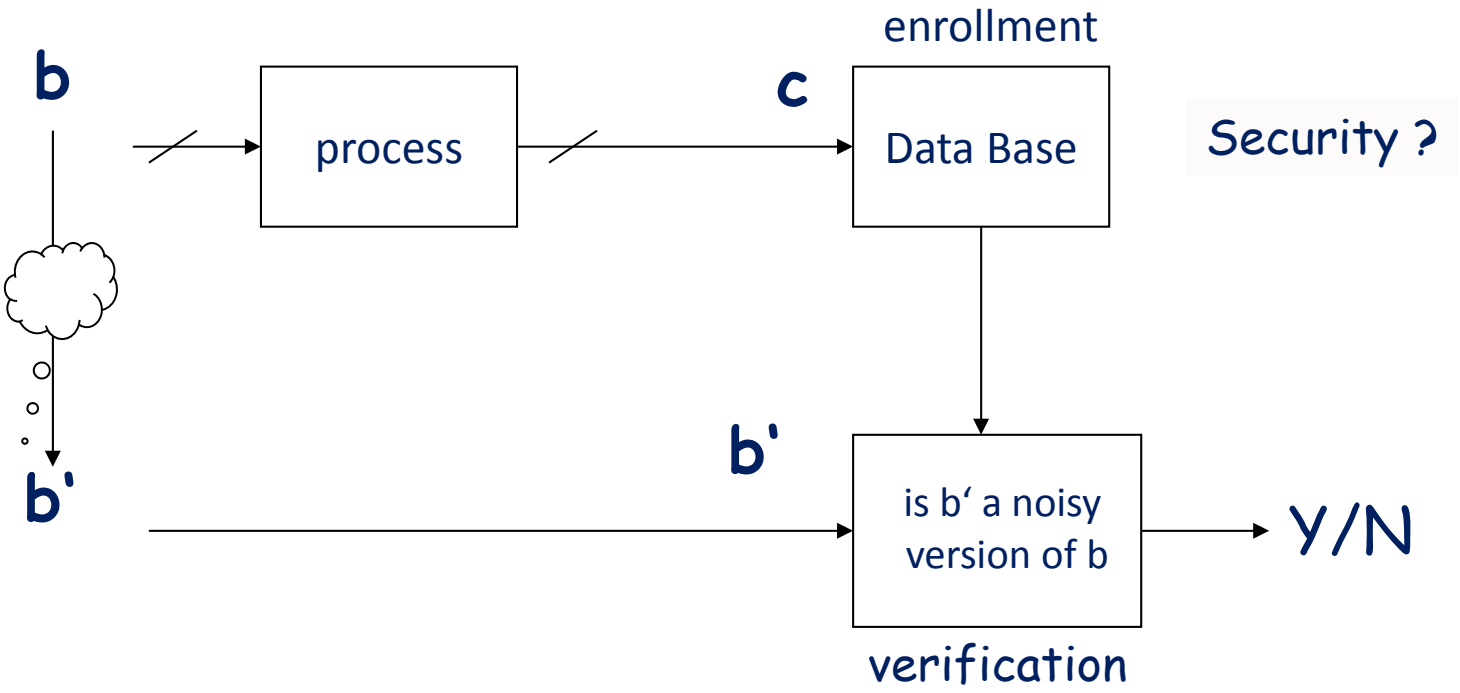
Example 1

Age 4 Age 6 Age 34 Age 35 Age 65

Example 2



Basic problem: aging introduces (permanent) errors

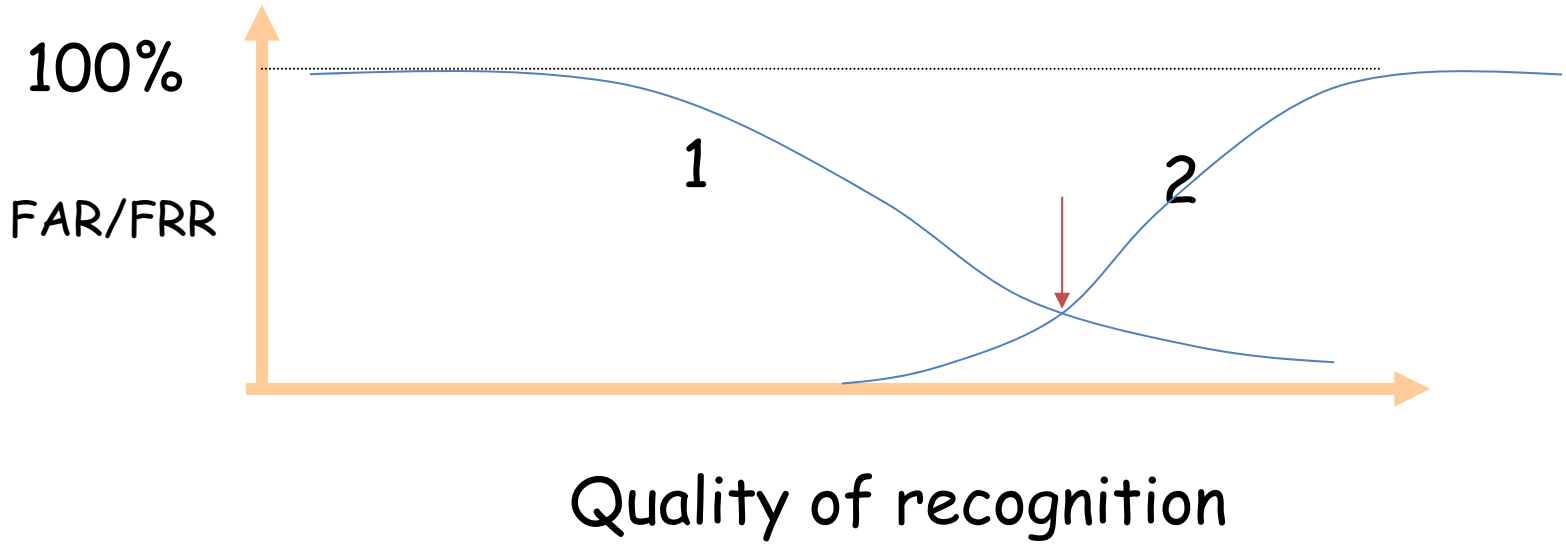


problem: how to do the processing and verification

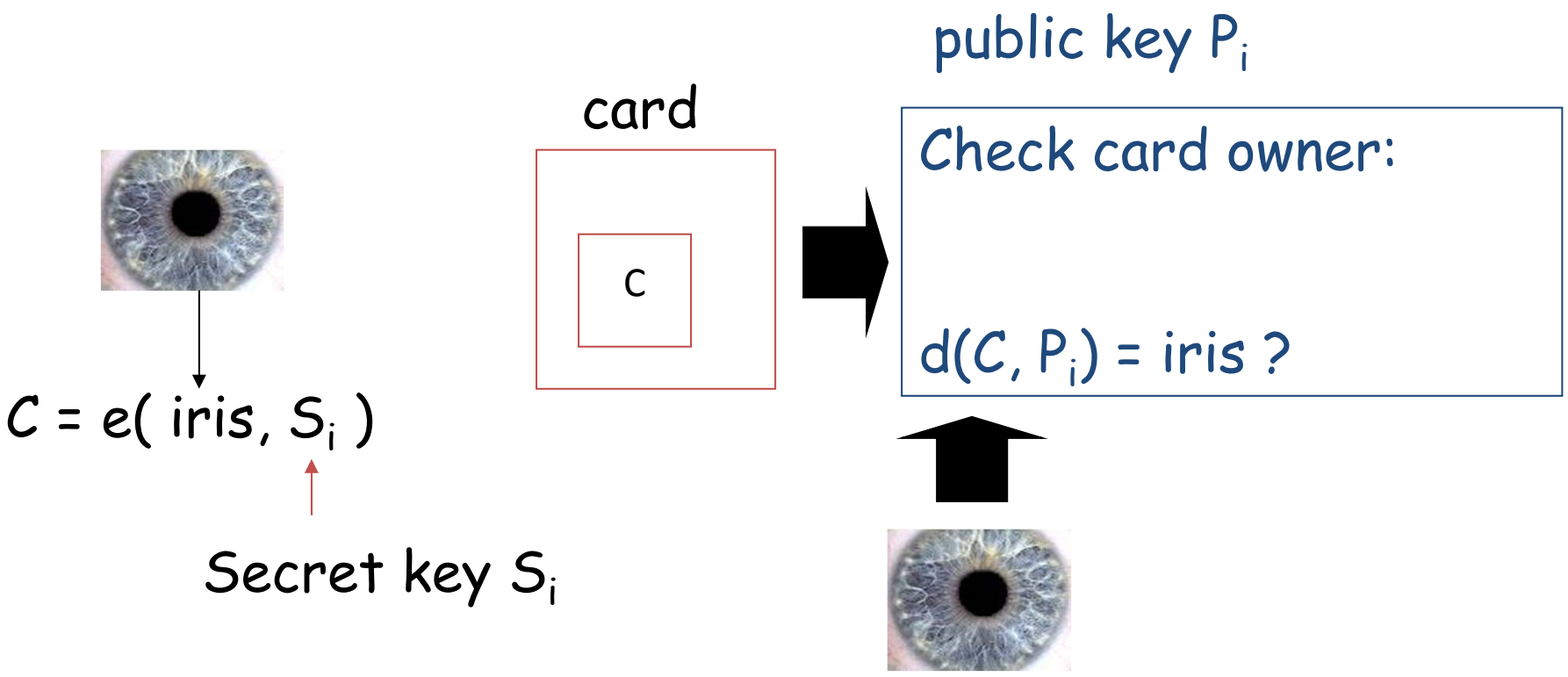
Biometrics, performance

Performance measures:

- 1. False acceptance rate (FAR) (imposter accepted)
- 2. False rejection rate (FRR) (legitimate match denied)



User identity check: example



Template Size

Biometric	Approx Template Size
Voice	70k – 80k
Face	84 bytes – 2k
Signature	500 bytes – 1000 bytes
Fingerprint	256 bytes – 1.2k
Hand Geometry	9 bytes
Iris	256 bytes – 512 bytes
Retina	96 bytes

The connection with information theory

- For perfect secrecy: the number of messages #(M) = #(M|C)
- System leakage: #(M)/#(M|C) ≥ 1

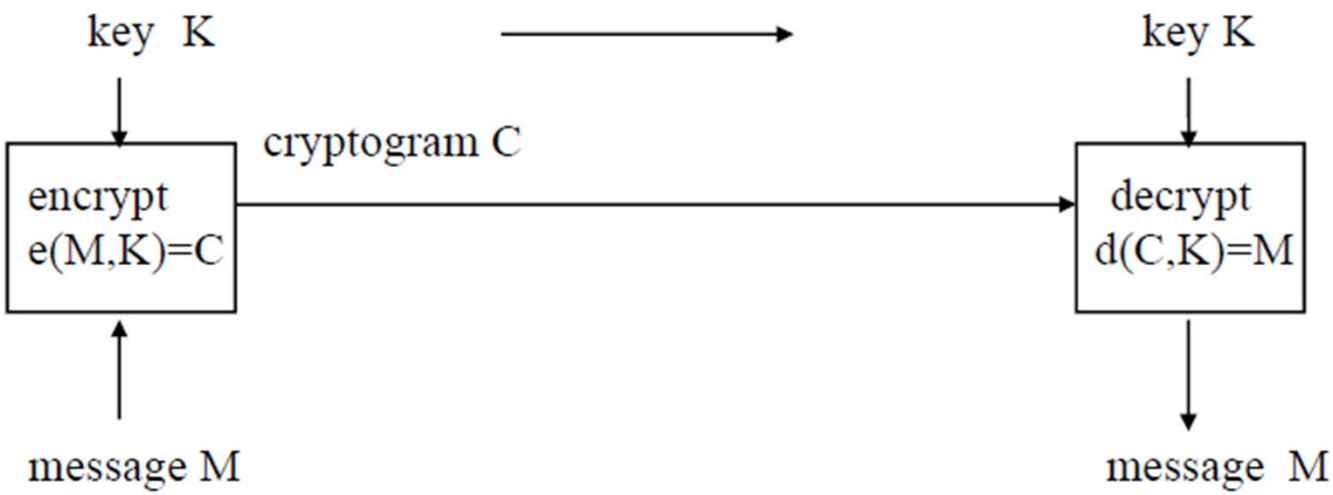


Figure 6.1 Schematic representation of the Shannon cipher model

Starting situation: intuitive analysis

For perfect secrecy: #(M) = #(M|C) = #(K|C) $\leq$ #(K)

C and M connected via unique key. Thus, M and C determine K

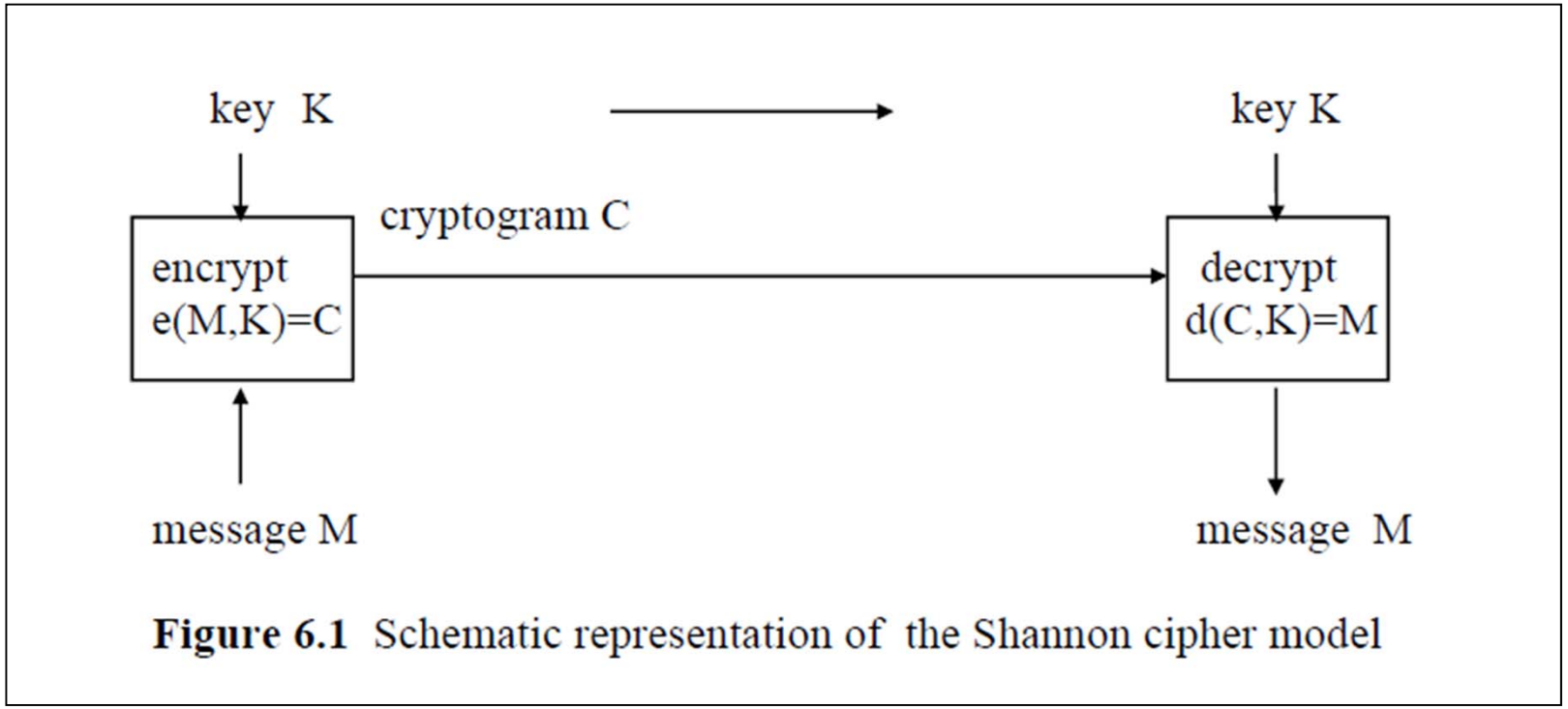


Figure 6.1 Schematic representation of the Shannon cipher model

Noisy key

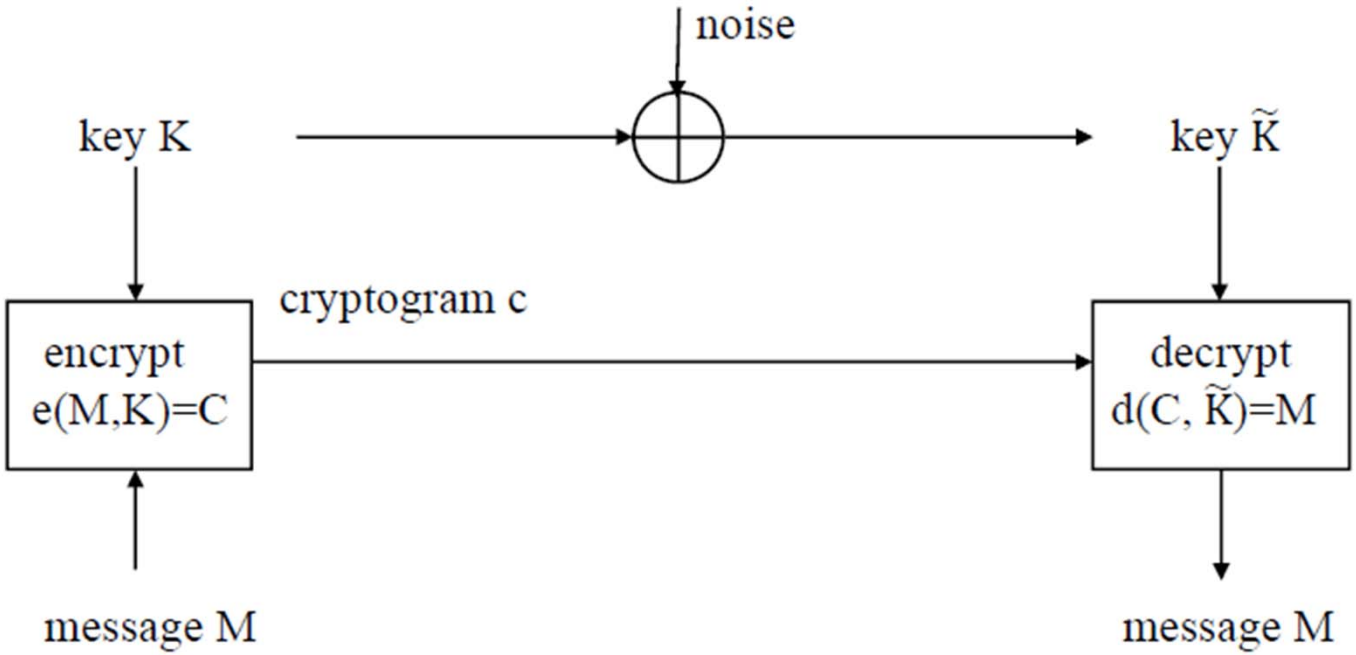


Figure 6.2 Schematic representation of the noisy cipher system

Noisy key

#(k‘) = number of noisy keys
#(k‘|k) = number of noisy keys given a key
#(k|c) = number of keys given a cipher

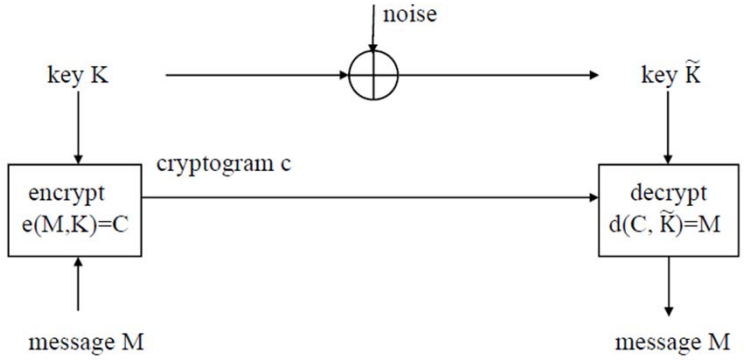


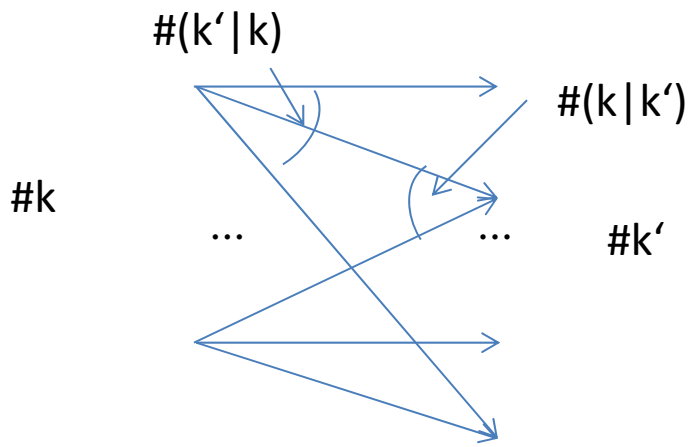
Figure 6.2 Schematic representation of the noisy cipher system

Let every key gives rise to a set of keys k‘|k => we assume the cardinality #(k‘|k) is fixed

Then, $\#(k|c) \times \#(k'|k) \leq \#(k')$.

Necessary condition, because if not true, there exists a key k‘ that originates from 2 or more different keys and thus incorrect decryption appears

Noisy key



Let #(k‘|k) be the number of noisy keys given a particular key (the same for all keys)
and the average number of keys given a noisy key is denoted as av(#(k|k‘))

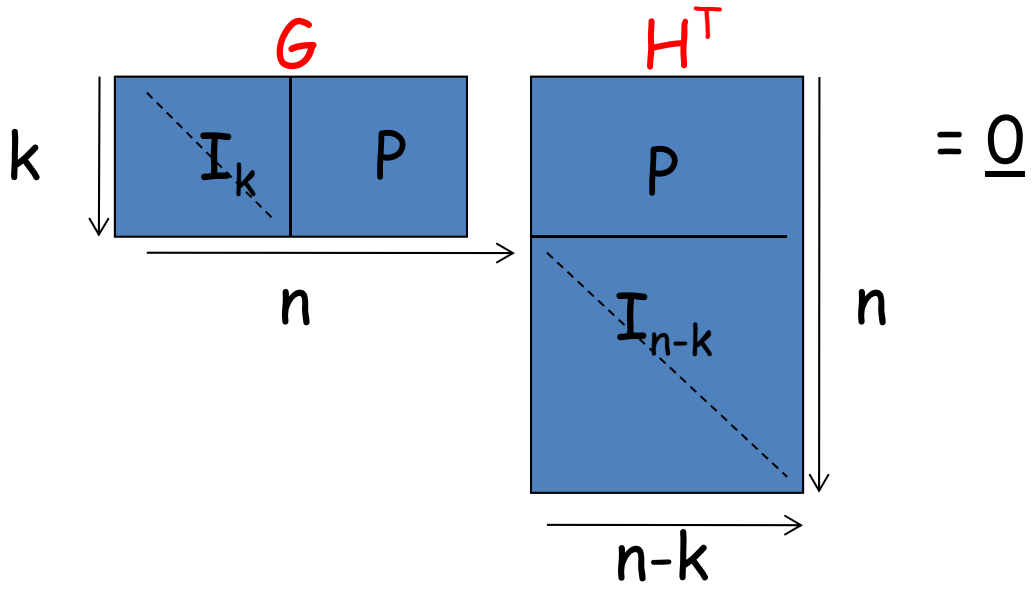
Then #(k) · #(k‘|k) = #(k‘) · av(#(k|k‘)) => the # of outgoing arrows = # incoming arrows

and thus for perfect secrecy: #(M) = #(M|c) = #(k|c) ≤ #(k‘) / #(k‘|k) = #(k) /av(#(k|k‘))

Conclusion: The noisy key gives a reduction in the maximum number of messages

idea: Use redundancy to correct errors in the Bio

Properties of a linear code: length n, k information digits
odd minimum distance d_{min}



property: $\underline{r}G = \underline{c}$

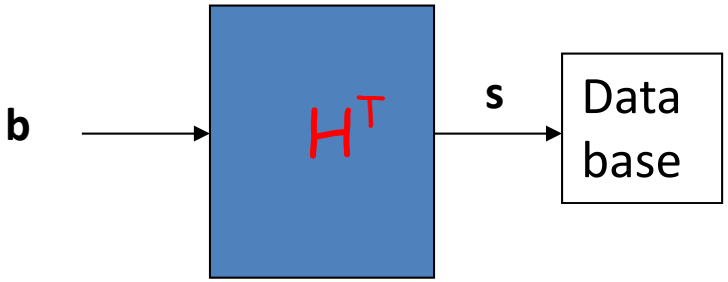
$\underline{c}H^T = \underline{0}$

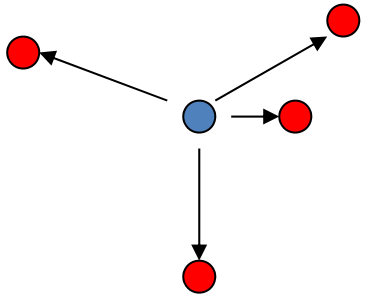
Property: let $e_1H^T = s_1$ and $e_2H^T = s_2$; $e_1 \neq e_2$

then $s_1 \neq s_2$ for $|e_1|$ and $|e_2| < d_{min}/2$ because…

Maximum Aposteriori Probability (MAP) receiver
(minimum error probability)

Given a channel





Attacker of DB: for every s, guess a particular b_i

- the best guess is the b_i for which $P(b_i \text{ stored as } s \mid s)$ is maximum

$$P(\text{correct} \mid s) = \max_b P(b \mid s)$$

$$\bar{P}(\text{correct}) = \sum_s P(s) \max_b P(b \mid s) = \sum_s \max_b P(s \mid b)P(b) \qquad \text{Bayes rule}$$

performance

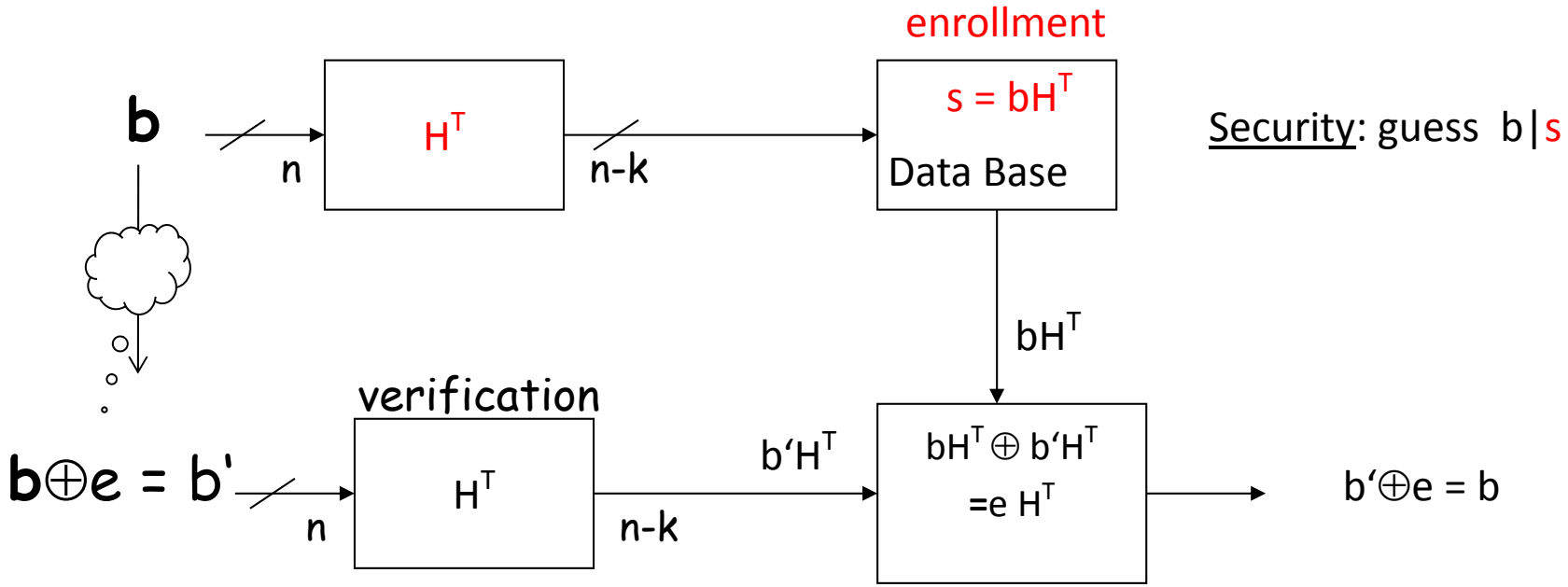
Minimum error propability guess (MAP)

- Guess $\underline{b} \rightarrow$ $\quad \bar{P}_{guess}(\text{correct}) \leq \max_{B} P(\underline{b}).$

- Guess $\underline{b}|\underline{s} \rightarrow$ $\quad \bar{P}_{guess}(\text{correct}|\underline{s}) = \sum_{\underline{s}\in S} P(\underline{s}) \max_{\underline{b}\in B} P(\underline{b}|\underline{s})$ $\quad \leq 2^{n-k} \max_{B} P(\underline{b}).$

We pay a price by using redundancy !

construct b from a noisy version b' and syndrome s



Conclusion:

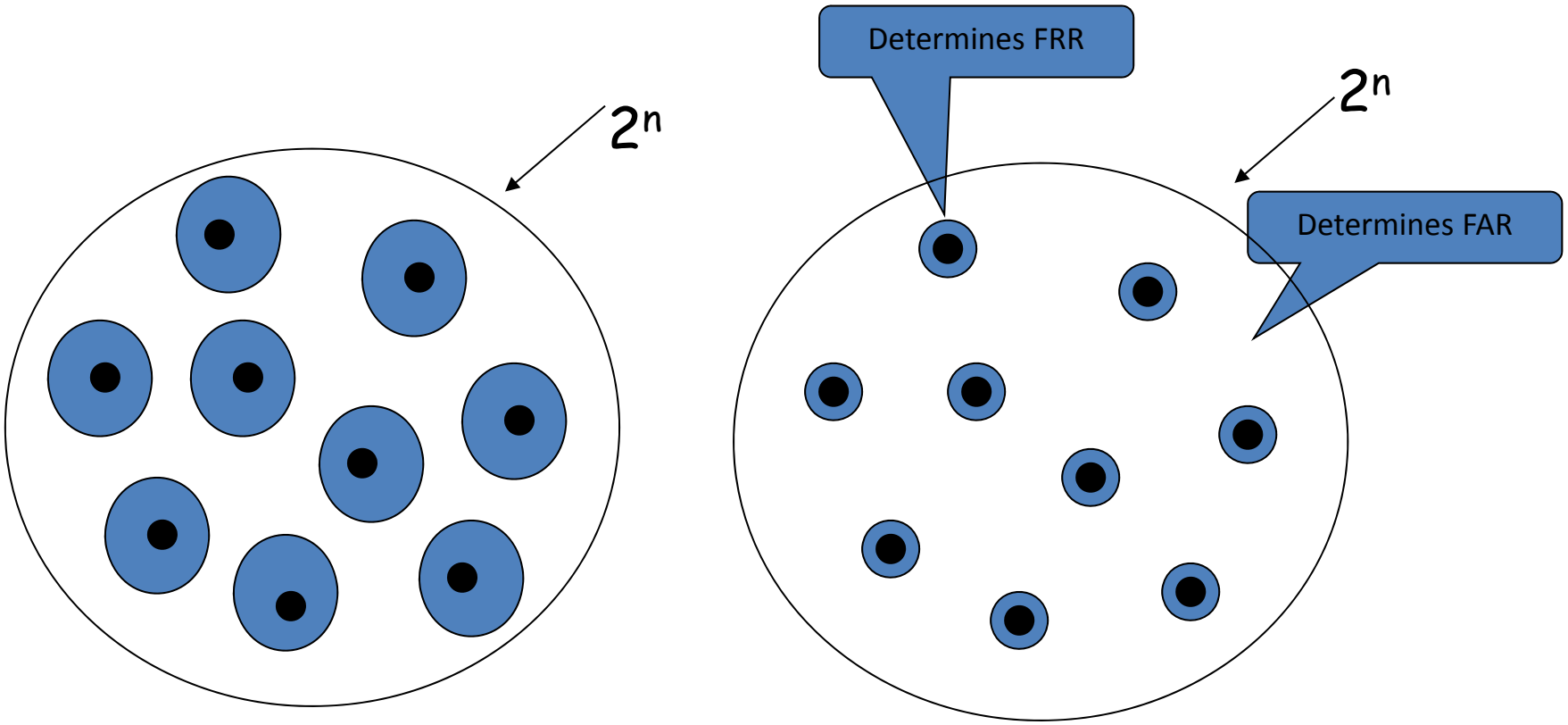
For k small:	good reconstruction, bad security
For k large:	bad reconstruction, good security

Example: BCH codes (bits) test for a valid syndrome

For binary BCH codes: n = 256, k = 224 bits, $d_{min} = 7$

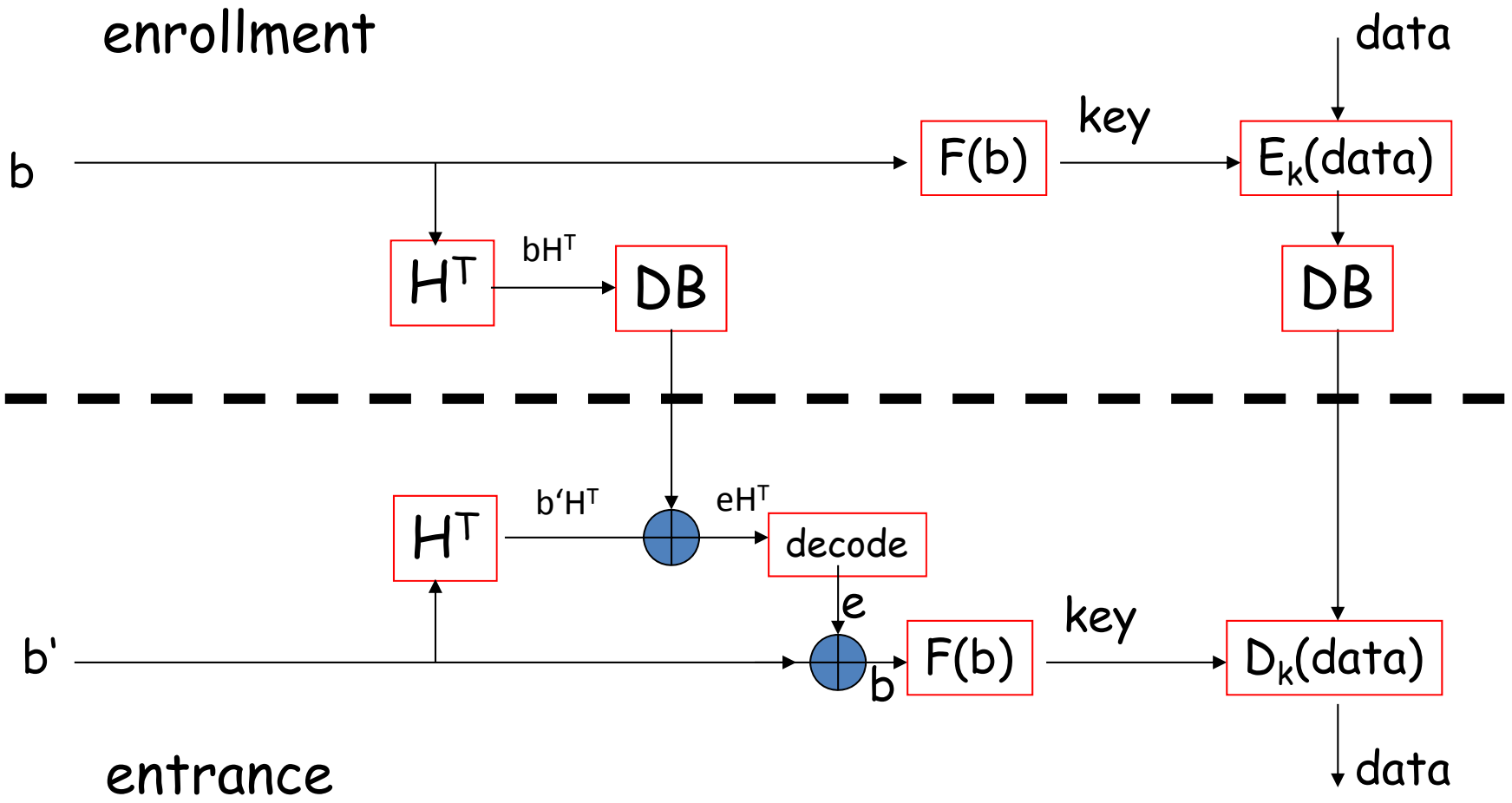
- False Rejection Rate = P(#errors ≥ 4) ≈ $(100p)^4$;
  too many differences
- False Acceptance Rate < 2^{-8}
  random vector insided decoding region
- Security: 2^{-224}

As a picture

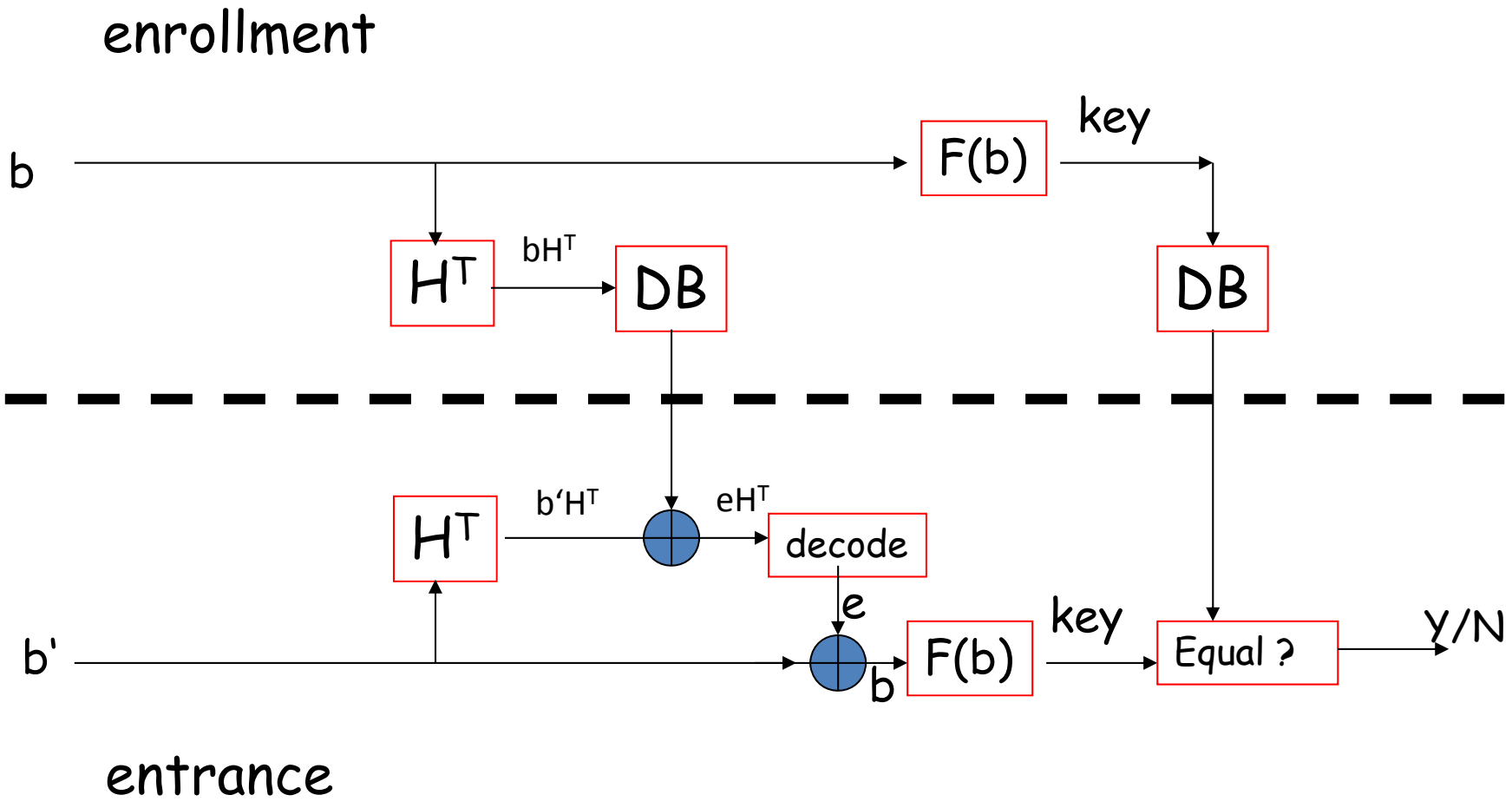


Number of codewords and length stays the same

It is time for an application

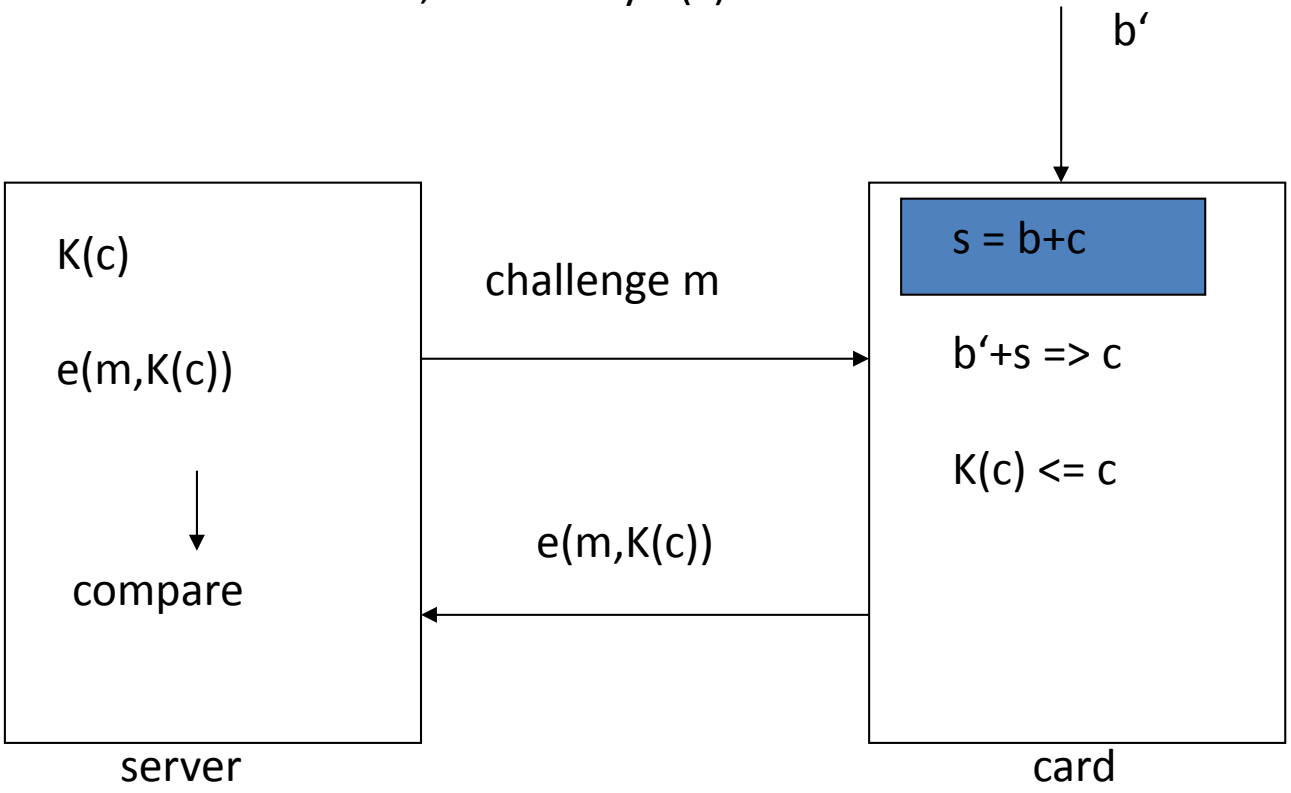


Another application

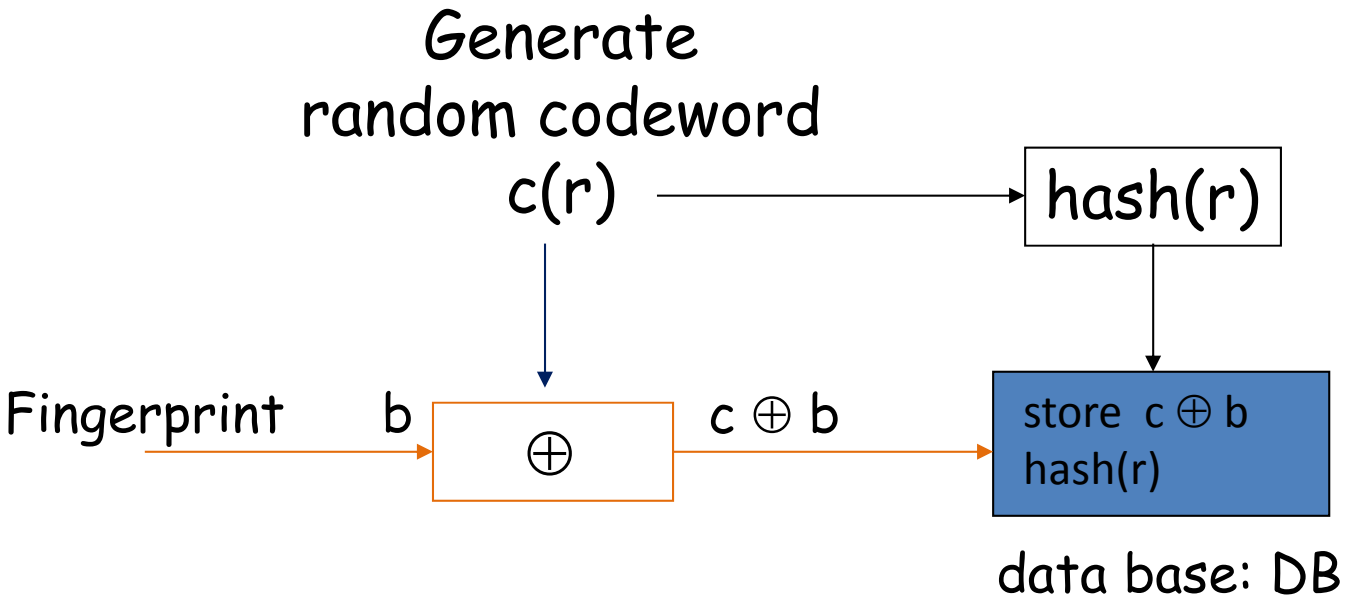


Challenge response

Enrollment: b + c = s; derive key K(c)



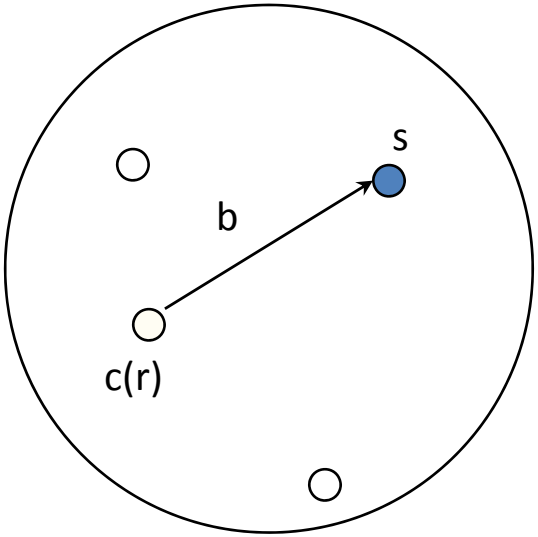
Another scheme: Enrollment



Condition: given $c \oplus b$ and hash(r)
it is hard to estimate b or c(r)

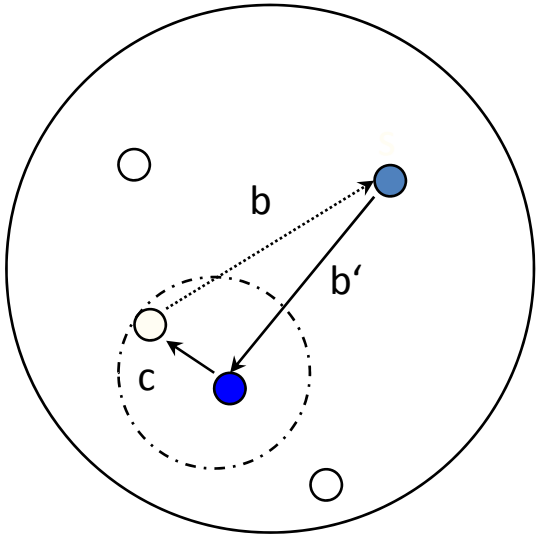
Idea: Juels-Wattenberg

Enrollment: b = fingerprint



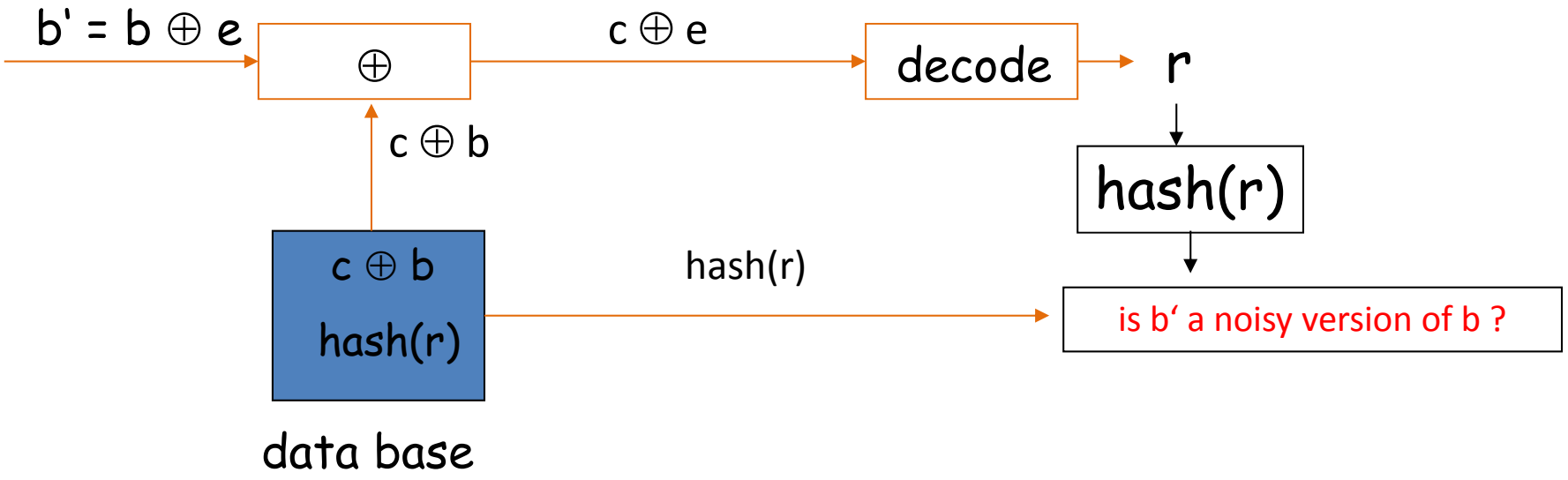
2^k Codewords c
- choose random r
- store s : s = c ⊕ b

Secure sketch: input b‘



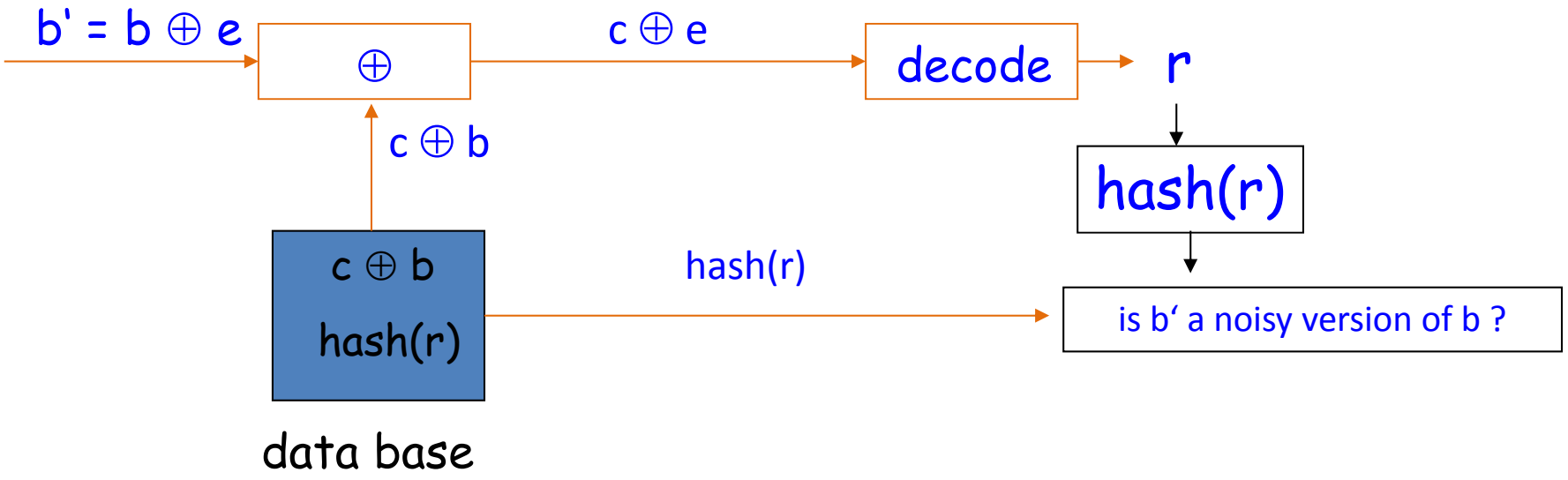
- decode c from s ⊕ b‘
- calculate s ⊕ c = b

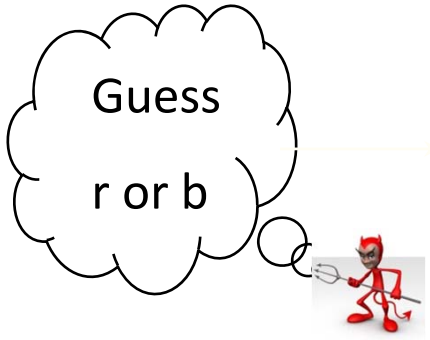
authentication



FRR: valid b' rejected; FAR: invalid b' accepted;

attacker



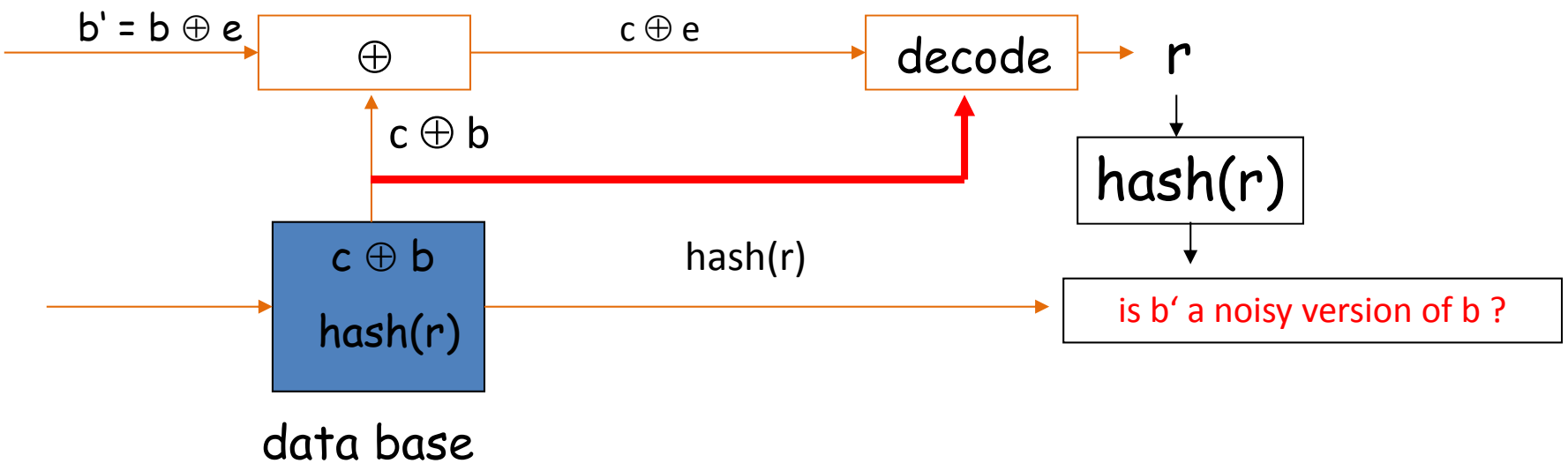


find b from $s \oplus c(r) = b$

or

find r from $s \oplus b = c(r)$

Improved legal detector



FRR: valid b' rejected; FAR: invalid b' accepted;