HMM: AUTHENTICATION THROUGH KEYSTROKE DYNAMICS

Michele Rossi
rossi@dei.unipd.it

Dept. of Information Engineering
University of Padova, IT

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Outline

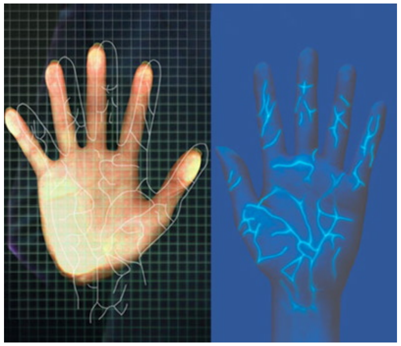
- Biometric authentication: introduction
- Keystroke dynamics for user authentication
- Example results

Biometric authentication

- An automatic method
  - identifies users *or* verifies their identity
  - Involves something one is *or* does

- Types of Biometrics
  - Physiological signal
  - Behavioural

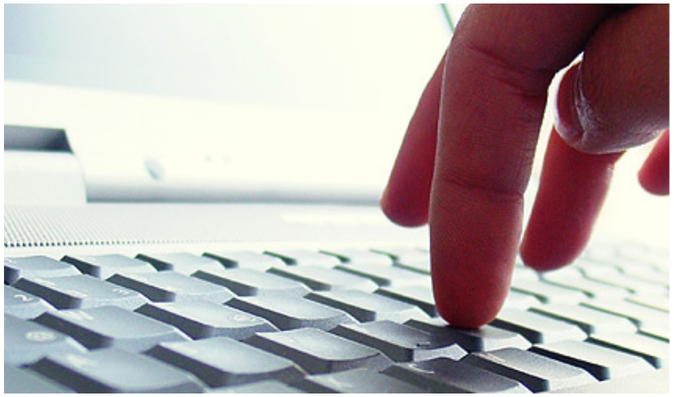
Biometric authentication

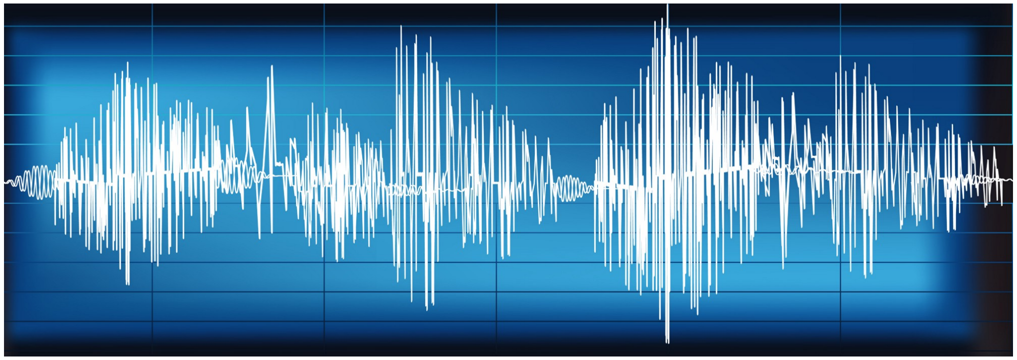


- Methods
  - Fingerprints
  - Iris, retinal scanning
  - Hand shape geometry
  - Blood vessel/vein pattern
  - Facial recognition
  - Voice pattern
  - DNA

Behavioral characteristics

- Hand written signatures
- Voice pattern
- Mouse movement dynamics
- Gait (way of walking)
- Keystroke dynamics





Keystroke dynamics patents

- [1] R. Bakis, D. Kanevsky, S.H. Maes, "Method and apparatus for recognizing identity of individuals employing synchronized biometrics", US Patent 6,219,639, Google Patents, 2001.
- [2] Vir V. Phoha, S. Phoha, A. Ray, S. S. Joshi, S. K. Vuyyuru, "Hidden markov model ("HMM")-based user authentication using keystroke dynamics", US 8136154 B2, 2012. (Published 12 Mar 2012, PennState and LouisianaState Universities, US)
- In this slideset we discuss the technique in [2]

Keystroke history

- Typing rhythms is an idea whose origin lies in the observation (made in 1897) that telegraph operators have distinctive patterns of keying messages over telegraph lines (behavioral biometrics)

- In keeping with these early observations, British radio interceptors, during World War II, identified German radio-telegraph operators by their "fist”, the personal style of tapping out a message

Keystroke verification: a quick taxonomy

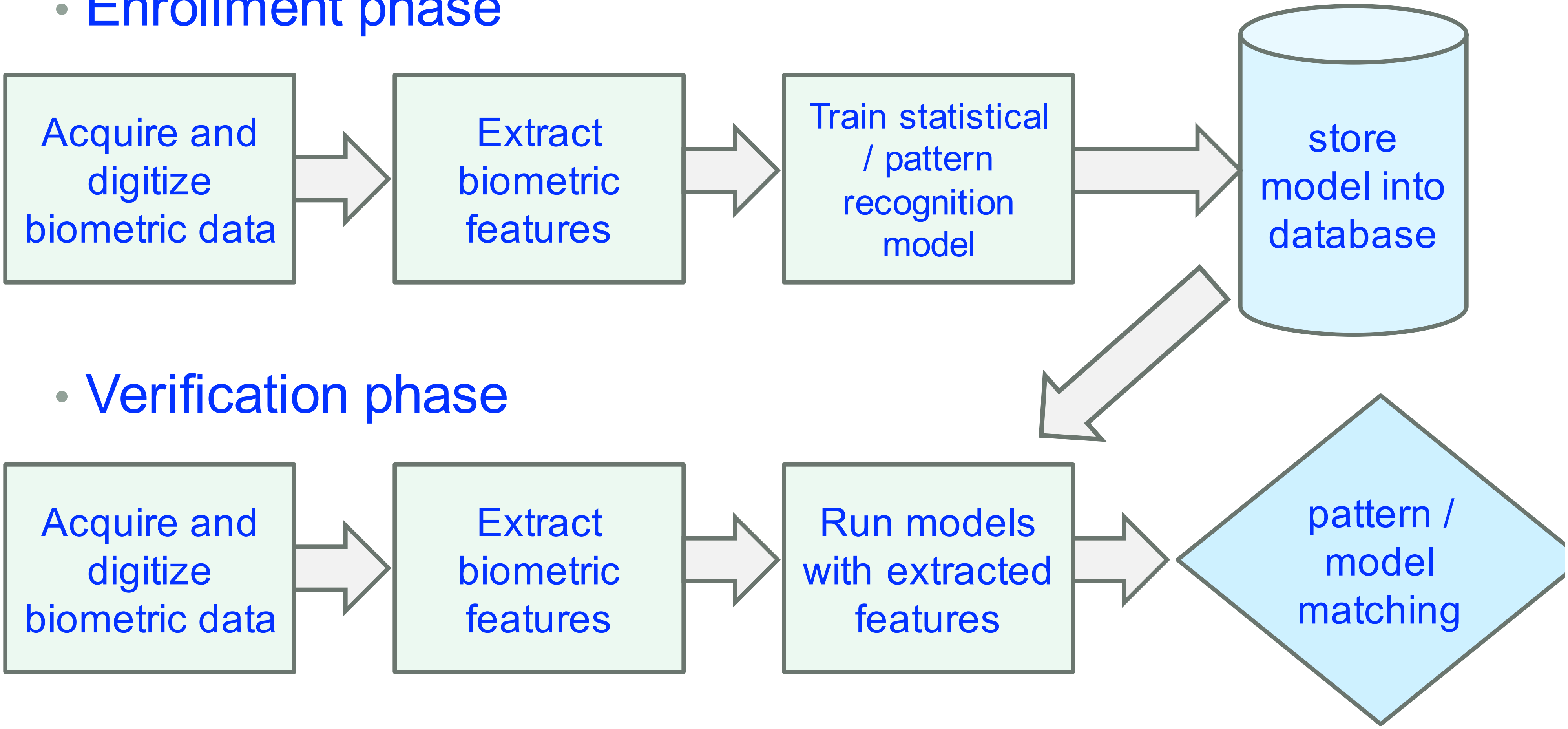
- A Behavioral measurement system aiming to identify users based on typing pattern / rhythms or attributes

- Static verification (fixed text mode)
  - Only based on password typing rhythm
  - Authentication only at login time

- Dynamic verification (free text mode)
  - pattern regardless of the typed text
  - A continuous or periodic monitoring (on-the-fly user authentication)
  - Not necessary to memorize a predefined text (username & passwd)

Biometric system

- Enrollment phase
- Verification phase



Acquire and digitize biometric data

Extract biometric features

Train statistical / pattern recognition model

store model into database

Acquire and digitize biometric data

Extract biometric features

Run models with extracted features

pattern / model matching

look for best match
(highest score)

Type of events & metrics

- U1 : **genuine, registered user**
- U2 : impostor, unregistered user

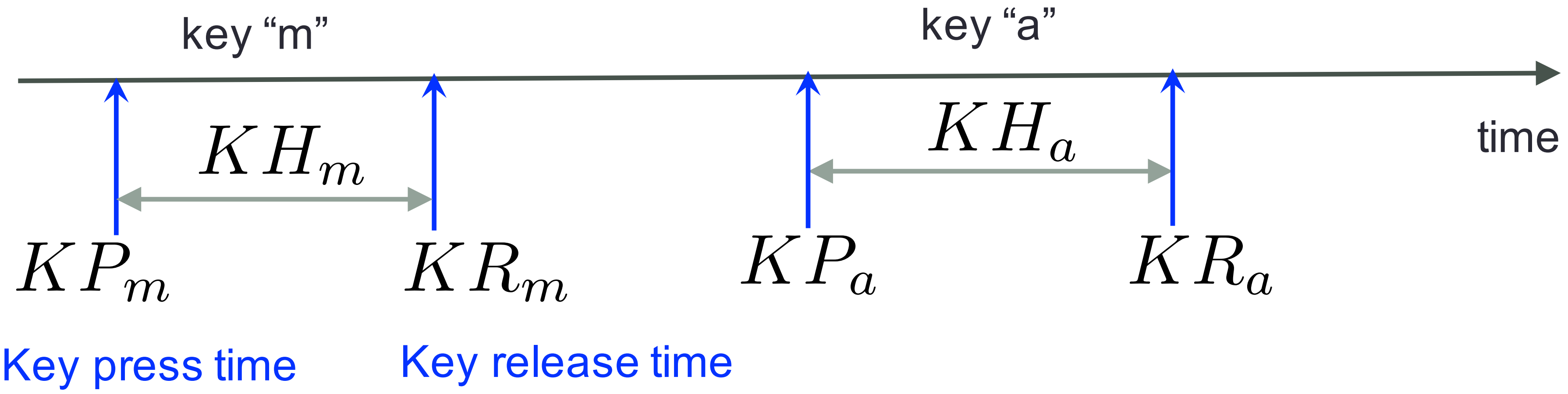
- Case 1: U1 claims to be U1 & gets accepted → **True Positive**
- Case 2: U1 claims to be U1 & gets rejected → False Reject
- Case 3: U2 claims to be U1 & gets accepted → False Accept
- Case 4: U2 claims to be U1 & gets rejected → **True Negative**

- A good authentication system must have:
  - a low False Acceptance Rate (FAR),
  - and a low False Rejection Rate (FRR)

- Note: False Reject = False Negative, False Accept = False Positive

Data features

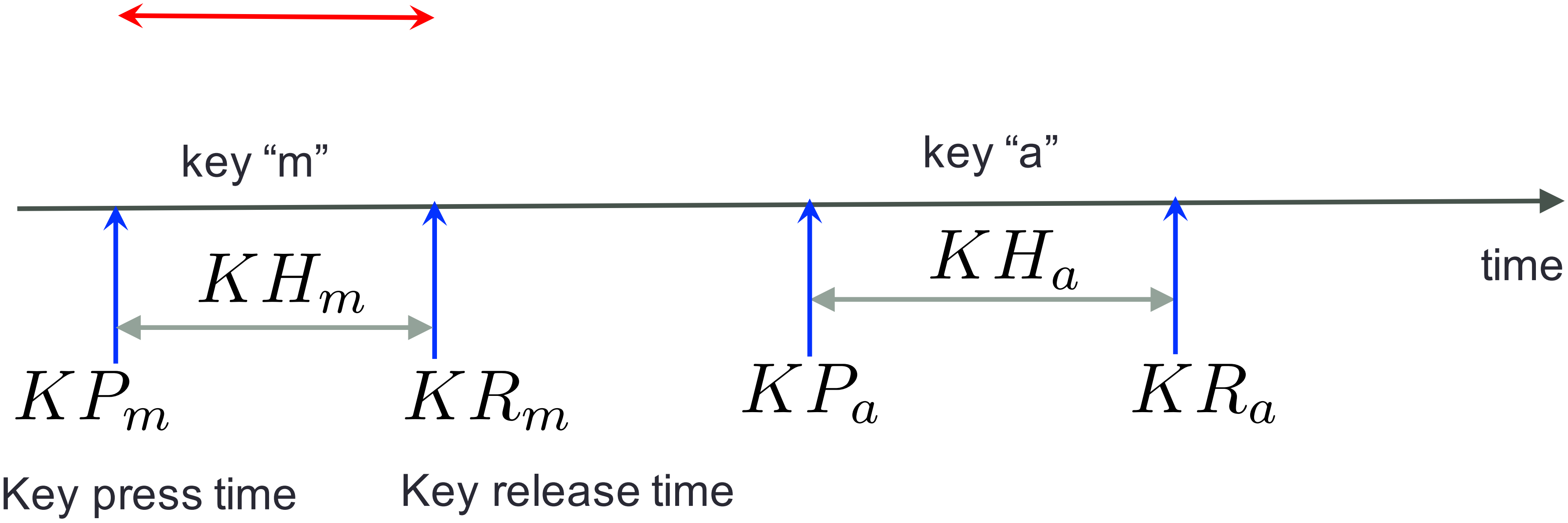
- For each keystroke pattern the following features are collected
  - (1) key hold time, (KH)
  - (2) key press latency, (KPL)
  - (3) key release latency, (KRL)
  - (4) key interval time (KI)
- E.g., keystroke pattern “ma”



Data features

- For each keystroke pattern the following features are collected
  - (1) key hold time

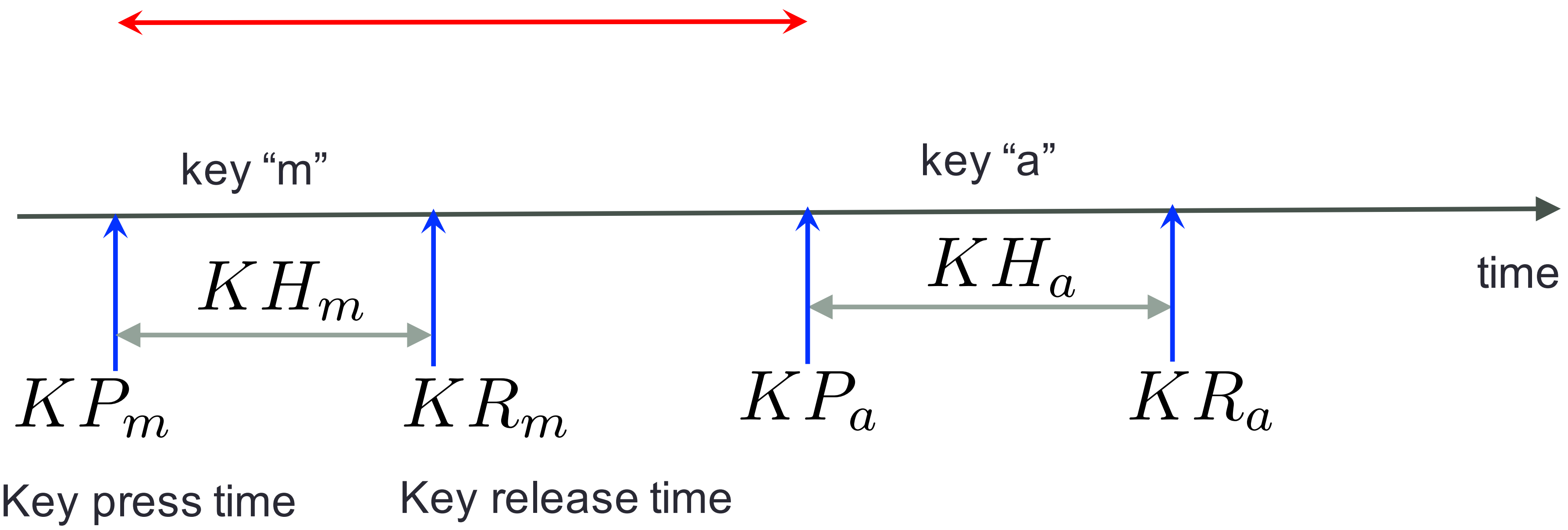
$$KH_m = KR_m - KP_m$$



Data features

- For each keystroke pattern the following features are collected
  - (2) key press latencies

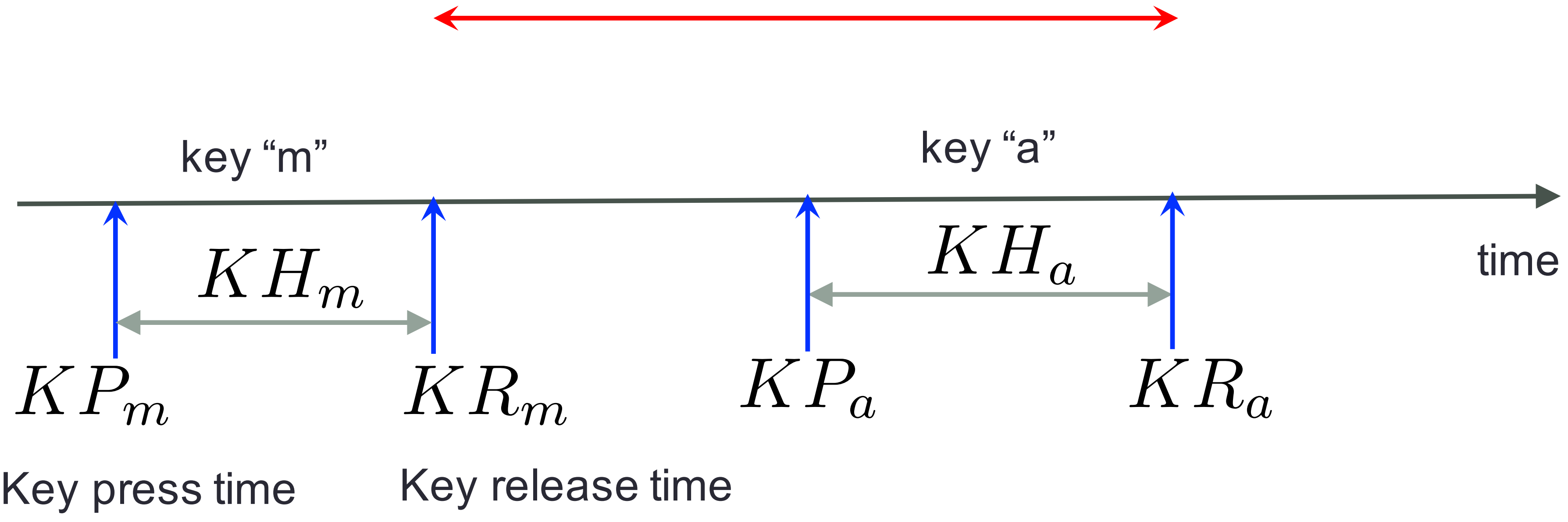
$$KPL_{ma} = KP_a - KP_m$$



Data features

- For each keystroke pattern the following features are collected
    - (3) key release latencies

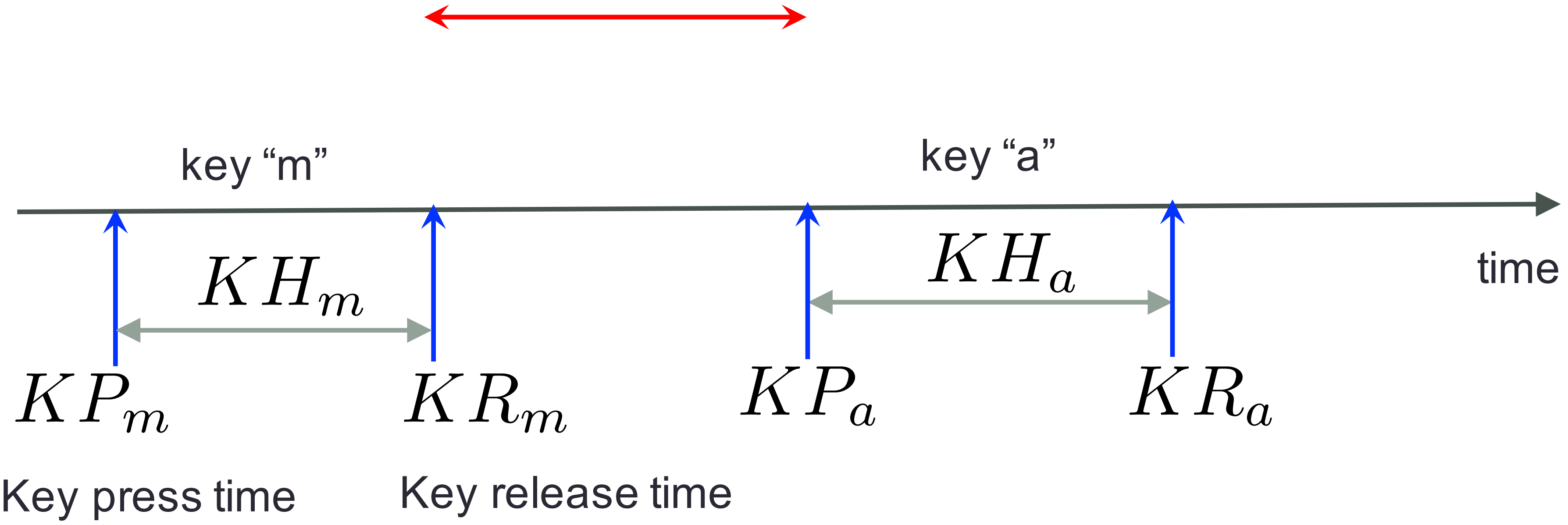
$$KRL_{ma} = KR_a - KR_m$$



Data features

- For each keystroke pattern the following features are collected
  - (4) key interval times

$$KI_{ma} = KP_a - KR_m$$



Feature vectors & dataset

- Features: previous studies indicate that Key holding times are the most representative feature [1][2]
- Dataset: 43 users provided a set of 9 keystroke patterns for the (same) sentence
  - "master of science in computer science" (37 characters)
  - Key holding times are used to form feature vectors
  - # of test patterns varies for each user from 0 to 102
  - A total of 873 patterns were collected

- [1] J. A. Robinson, V. M. Liang, J. A. Michael Chambers and C.L. Mackenzie, Computer user verification using login string keystroke dynamics. *IEEE Transactions on Systems, Man and Cybernetics* - Part A: Systems and Humans, 28 (2). 236-241, 1998.
- [2] M. S. Obaidat and B. Sadoun, Verification of Computer Users Using Keystroke Dynamics. *IEEE Transactions on Systems, Man, and Cybernetics* - Part B: Cybernetics, 27 (2). 261-269, 1997.

The HMM model

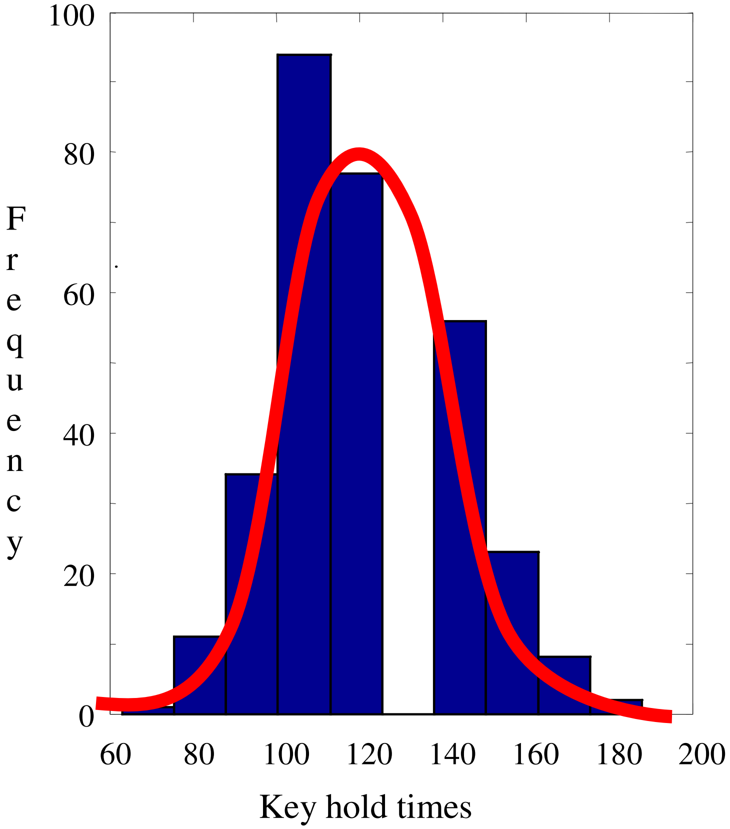
- Is a five-tuple: $\boldsymbol{\lambda} = \{\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{A}, \boldsymbol{\pi}, \boldsymbol{\Phi}\}$

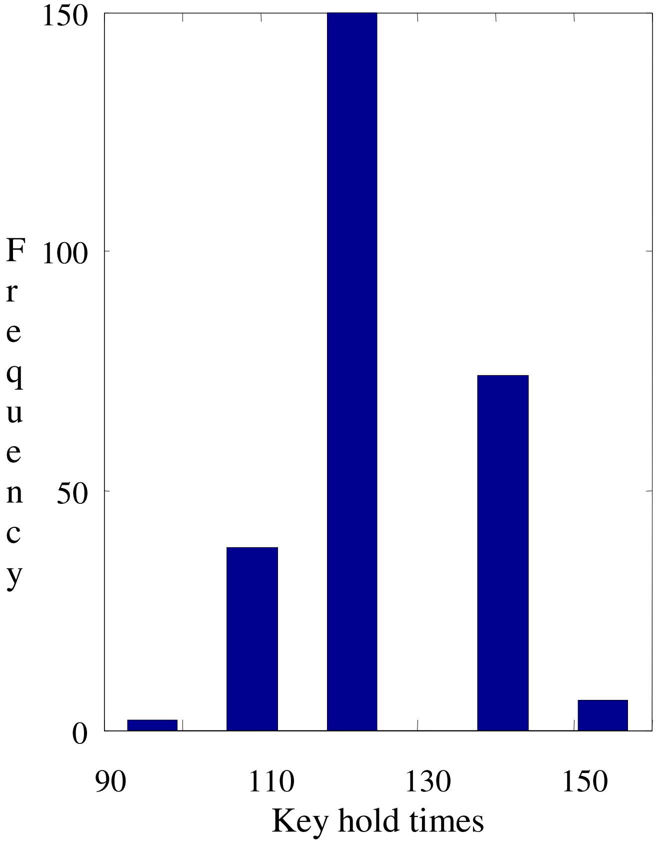
- Key hold dynamics of a string
  - Considered as “energy levels”
  - Sequence of “energy levels” **X** maps into a sequence of hidden states **Z**

- Time is discrete
  - New timestep each time a new key is stroked
  - Each letter corresponds to an “energy level”
    - “energy level” represents the corresponding key hold time

Observation model

- PDF of observation (hold time) x_t at time t
  - Given that hidden state z is “j”
  - Gaussian model is commonly used: $p(x_t|z_j,\phi_j) = G(x_t|\mu_j,\sigma_j)$



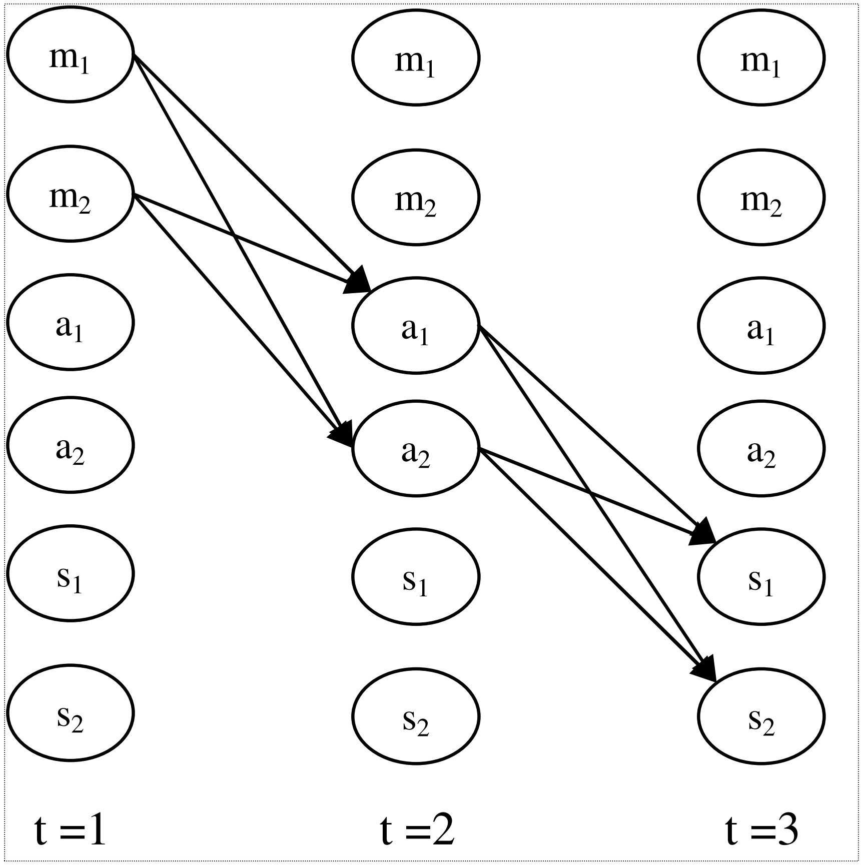


Hidden state sequence (example 1/3)

- The key sequence (e.g., password) is fixed
  - E.g., take the word “m” - “a” - “s”
  - The user is forced to go through these three letters…
  - …in this specific order

- What is not fixed is the “energy” associated with each letter
  - This energy in our model is the “hold time”
  - So we track the hold times sequence as the user types the password
  - This sequence should be characteristic of a specific user

Hidden state sequence (example 2/3)

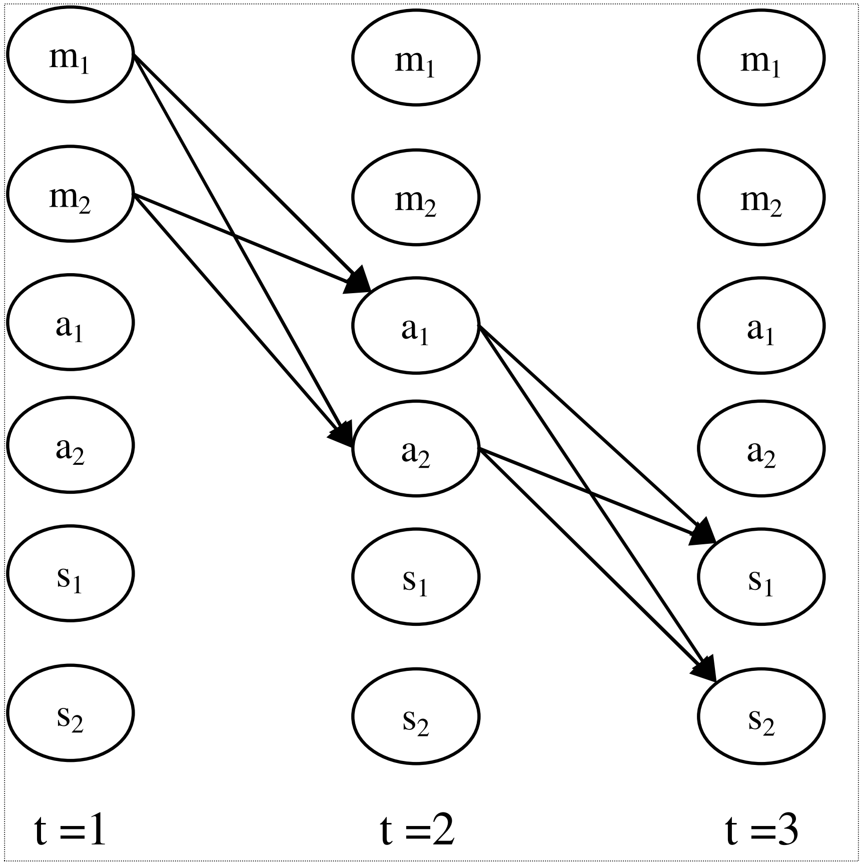


$$A = \begin{array}{c|cccccc} & m_1 & m_2 & a_1 & a_2 & s_1 & s_2 \\ \hline m_1 & 0 & 0 & 0.66 & 0.34 & 0 & 0 \\ m_2 & 0 & 0 & 0.80 & 0.20 & 0 & 0 \\ a_1 & 0 & 0 & 0 & 0 & 0.70 & 0.30 \\ a_2 & 0 & 0 & 0 & 0 & 0.58 & 0.42 \\ s_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_2 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$\pi = \begin{array}{c|c} m_1 & 0.58 \\ m_2 & 0.42 \\ a_1 & 0 \\ a_2 & 0 \\ s_1 & 0 \\ s_2 & 0 \end{array}$$

Input sequence is "m" - "a" - "s". At each time t (key stroke), in this example, only three letters are possible. Two sub-states are tracked for each letter. Each sub-state j has specific values for mean μ_j and variance σ_j (related to their hold times)

Hidden state sequence (example 3/3)



$$
A = \begin{array}{c|cccccc}
 & m_1 & m_2 & a_1 & a_2 & s_1 & s_2 \\
\hline
m_1 & 0 & 0 & 0.66 & 0.34 & 0 & 0 \\
m_2 & 0 & 0 & 0.80 & 0.20 & 0 & 0 \\
a_1 & 0 & 0 & 0 & 0 & 0.70 & 0.30 \\
a_2 & 0 & 0 & 0 & 0 & 0.58 & 0.42 \\
s_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_2 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

$$
\pi = \begin{array}{c|c}
m_1 & 0.58 \\
m_2 & 0.42 \\
a_1 & 0 \\
a_2 & 0 \\
s_1 & 0 \\
s_2 & 0
\end{array}
$$

First letter is forced to be an “m”. This means that steady state probability of starting in “a” and “s” are both zero. The unit probability is split across the two states for key “m” (e.g., longer and shorter version of it)

Observation model

- Initial model estimates
  - Time steps are t=1,…,37 (37 characters in the string) with T=37
  - There are *6 example patterns for the same string*
  - For every time t (character), empirical **mean** and **standard deviation** are estimated from the 6 example patterns, looking at the *key holding times* of the character typed at time t:

$$\mu_t, \sigma_t$$

  - A number of hidden states is then defined, each with a specific mean and standard deviation. If **five sub-states** s=1,2,…, 5 are tracked for each (letter) time t, we take their parameters (mean and stdev) equal to:

$$(\underset{\substack{\longleftrightarrow \\ \text{mean} \\ \mu_{ts}}}{\mu_t - 2\sigma_t}, \underset{\substack{\leftrightarrow \\ \text{stdev} \\ \sigma_{ts}}}{\sigma_t}),\ (\mu_t - \sigma_t, \sigma_t),\ (\mu_t, \sigma_t),\ (\mu_t + \sigma_t, \sigma_t),\ (\mu_t + 2\sigma_t, \sigma_t)$$

Structure of matrix **A**

- T=37 time steps
- For each time step t
  - A specific subset of states is tracked → N_{sc} sub-states, corresponding to the character typed in that time step
- Next time step t+1
  - The system is constrained to evolve towards a specific sub-set of states (still N_{sc} states), according to the values of mean and standard deviation that were measured during the training in step t+1
  - The number of possible transitions from time t to time t+1 is N_{sc} x N_{sc}

- The total number of transitions (a "path" through the states) is T x N_{sc} x N_{sc}
- Transition matrix A has a "3D structure" – $\boldsymbol{A} = \{a_{trs}\}$
  - Time index t (current time step, T states)
  - Index s (N_{sc}) is current character sub-state and index r (N_{sc}) is the next one

Observation PDFs

- Let us consider time t=1,…,T
  - Gaussian PDF associated with observation x_t
  - when the system is in the s-th sub-state of the t-th character (time step):

$$p(x_t|s) = G(x_t|\mu_{ts}, \sigma_{ts})\ ,\ s = 1, \ldots, N_{sc}$$

HMM training - forward recursion

- 1) initialization:

$$\alpha_1(r) = \pi_r G(x_1|\mu_{1r}, \sigma_{1r}) \, , \, r = 1, \ldots, N_{sc}$$

- 2) Forward recursion (from t to t+1):

$$\alpha_{t+1}(s) = \left[\sum_{r=1}^{N_{sc}} \alpha_t(r) a_{trs} \right] p(x_{t+1}|s) \, , s = 1, \ldots, N_{sc}$$

$$, t = 1, \ldots, T - 1$$

- 3) Termination, the likelihood:

$$P(\boldsymbol{X}|\lambda) = \sum_{s=1}^{N_{sc}} \alpha_T(s)$$

(T is the final time)

HMM training - backward recursion

- 1) initialization (last time step T):

$$\beta_T(s) = 1\,,\; s = 1, \ldots, N_{sc}$$

- 2) Backward recursion (from t+1 back to t):

$$\beta_t(r) = \sum_{s=1}^{N_{sc}} a_{trs} p(x_{t+1}|s) \beta_{t+1}(s)\,, r = 1, \ldots, N_{sc}$$

$$, \; t = 1, \ldots, T-1$$

HMM training approach

- Results into a set of models, one for each registered user:

$$\{\lambda_1, \lambda_2, \ldots, \lambda_U\}$$

- Multiple (short) training sequences are available for each user
  - In place of a single long training sequence
  - This means that the standard algorithm has to be modified
  - A popular re-estimation approach has been proposed by [Rabiner89]

- [Rabiner89] Lawrence R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” *Proceedings of the IEEE*, Vol. 77, No. 2, February 1989.

Summary of quantities γ_t(i)

- Let i be the system state at time step t
- Let N_{sc} be the number of states at time t

$$\gamma_t(i) = P(s_t = i|\boldsymbol{X}, \lambda)$$

- Is the probability that the system at time t is in state i, given the observations x_1, x_2, …, x_T and the model λ
- From the theory, we know that:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(\boldsymbol{X}|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N_{sc}} \alpha_t(i)\beta_t(i)}$$

Summary of quantities $\xi_t(i,j)$

- Let i and j be the system state at time step t and t+1:

$$\xi_t(i,j) = P(s_t = i, s_{t+1} = j | \boldsymbol{X}, \lambda)$$

- can be computed as:

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{tij} p(x_{t+1}|j) \beta_{t+1}(j)}{P(\boldsymbol{X}|\lambda)} =$$

$$= \frac{\alpha_t(i) a_{tij} p(x_{t+1}|j) \beta_{t+1}(j)}{\sum_{i=1}^{N_{sc}} \sum_{j=1}^{N_{sc}} \alpha_t(i) a_{tij} p(x_{t+1}|j) \beta_{t+1}(j)} =$$

$$= \frac{\alpha_t(i) a_{tij} p(x_{t+1}|j) \beta_{t+1}(j)}{\sum_{i=1}^{N_{sc}} \alpha_t(i) \beta_t(i)}$$

Summary of quantities a$_{tij}$

- Note that, the transition probabilities are obtained as:

$$a_{tij} = \frac{\text{expected no. of transitions from } i \text{ to } j \text{ at time } t}{\text{expected no. of transitions from state } i} =$$

$$= \frac{\xi_t(i,j)}{\gamma_t(i)}$$

- The normalizing term

$$p(\boldsymbol{X}|\lambda) = \sum_{i=1}^{N_{sc}} \alpha_t(i)\beta_t(i)$$

cancels out, as both the numerator and denominator have it

Rabiner’s re-estimation (1/4)

- Let K be the number of available short test observation sequences for a user

$$\{\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \ldots, \boldsymbol{X}^{(K)}\}$$

- where:

$$\boldsymbol{X}^{(k)} = \{x_1^{(k)}, x_2^{(k)}, \ldots, x_T^{(k)}\}$$

- is the k-th observation sequence

- Notation: for any variable, $x^{(k)}$, refers to the value the variable takes for the *k-th model*

Rabiner’s re-estimation (2/4)

- Steady-state probabilities:

new global model parameters

$$\boxed{\pi_r} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \boxed{\gamma_1^{(k)}(r)}}{\sum_{k=1}^{K} \frac{1}{P_k}}$$

estimate from short sequence k

- where:

$$P_k \triangleq P(\boldsymbol{X}^{(k)}|\lambda)$$

- is the likelihood of the k-th sequence
- Rationale: the sequences that are less fit to the current model (have a small P_k) are those with the highest assigned weight → the new model will be updated *mostly* based on them

Rabiner’s re-estimation (3/4)

- Update formula for transition probabilities:

new estimates

$$a_{trs} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \xi_t^{(k)}(r,s)}{\sum_{k=1}^{K} \frac{1}{P_k} \gamma_t^{(k)}(r)} =$$

old estimates

$$= \frac{\sum_{k=1}^{K} \frac{1}{P_k} \left(\alpha_t^{(k)}(r) a_{trs} p(x_{t+1}|s) \beta_{t+1}^{(k)}(s) \right)}{\sum_{k=1}^{K} \frac{1}{P_k} \left(\alpha_t^{(k)}(r) \beta_t^{(k)}(r) \right)}$$

normalization factors are the same and cancel out

Rabiner’s re-estimation (4/4)

- Update formulas for mean and standard deviation:

$$\mu_{ts} = \frac{\sum_{k=1}^{K} \gamma_t^{(k)}(s) x_t^{(k)}}{\sum_{k=1}^{K} \gamma_t^{(k)}(s)}$$

average holding time for state s at time t: is obtained as the weighted average of the holding time for the *k-th pattern* at *time t*, normalized wrt the total prob of being in state s (over all models)

$$\sigma_{ts} = \frac{\sum_{k=1}^{K} \gamma_t^{(k)}(s)(x_t^{(k)} - \mu_{ts})^2}{\sum_{k=1}^{K} \gamma_t^{(k)}(s)}$$

same concept applies to the variance

Step 1 – best model selection

- Authentication, a test sequence is obtained from a user who claims to be the registered user with identiry (*username*) “cu”:

$$\tilde{\boldsymbol{X}} = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_T\}$$

- All U models are assessed against this sequence
  - Running the forward procedure and
  - Computing the likelihood

$$P(\tilde{\boldsymbol{X}}|\lambda_u)\,,\; u = 1, \ldots, U$$

  - The model leading to the highest likelihood is selected

$$u^* = \arg\max_u P(\tilde{\boldsymbol{X}}|\lambda_u)$$

Step 2 – verification

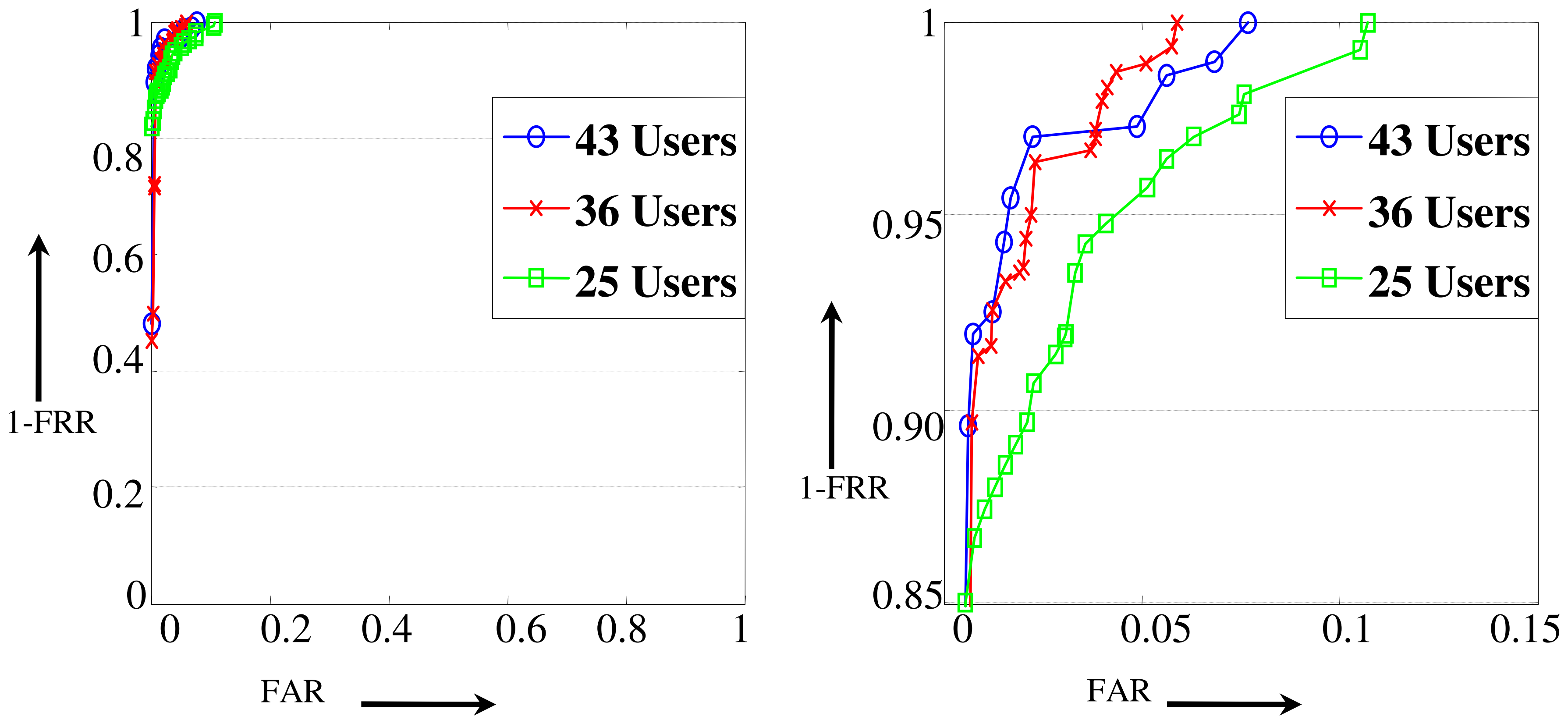
- The claimed user “cu” is accepted if the following conditions are both verified:

$$\begin{cases} \dfrac{P(\tilde{\boldsymbol{X}}|\lambda_{cu})}{P(\tilde{\boldsymbol{X}}|\lambda_{u^*})} \geq \theta_1 \\[2em] P(\tilde{\boldsymbol{X}}|\lambda_{cu}) \geq \theta_2 \end{cases}$$

1. Likelihood of claimed user and winner are sufficiently close
2. Likelihood of claimed user is sufficiently high

- The two thresholds can be tuned to obtain the desired performance in terms of FAR and FRR

Receiver Operating Characteristic (ROC)



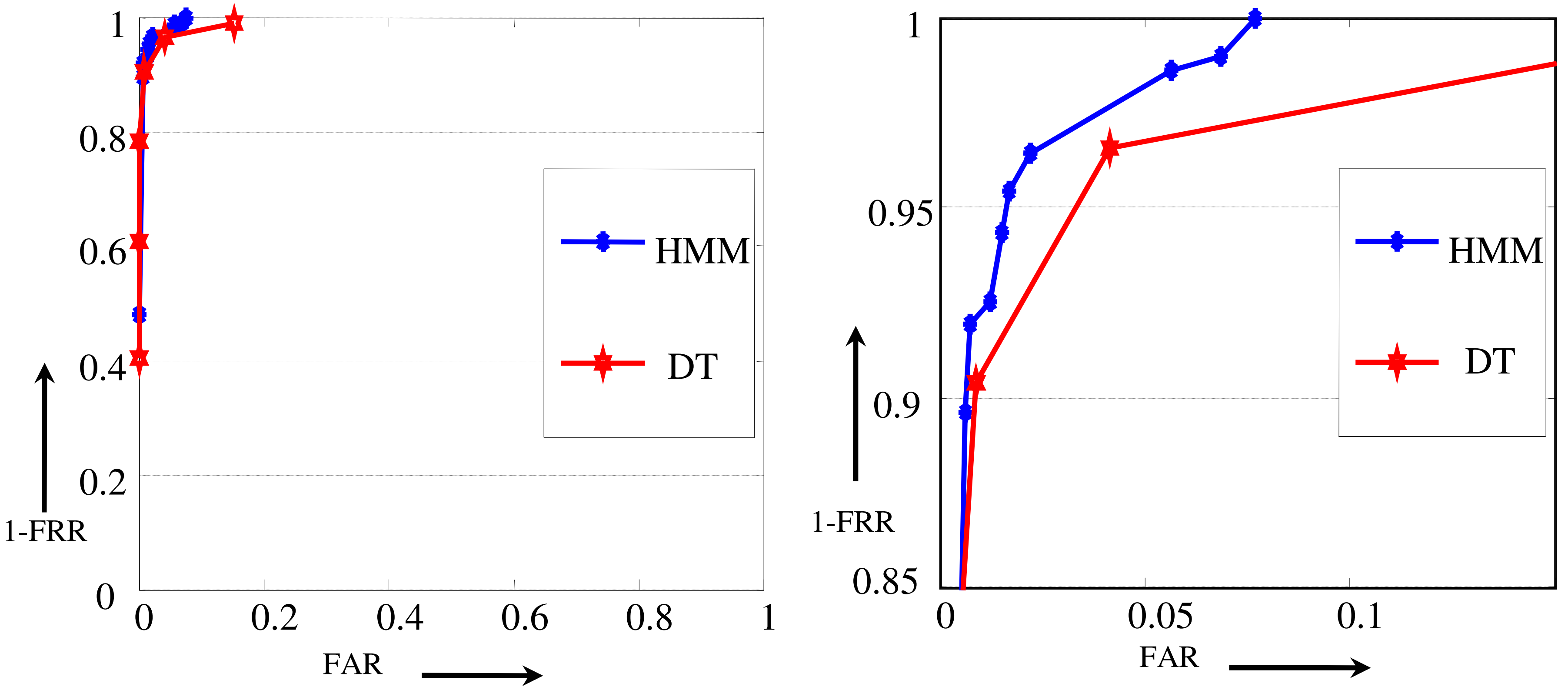
- Test data of 873 login attempts, three experiments:

  (1) 43 registered users & 0 unregistered,

  (2) 36 registered users & 7 unregistered,

  (3) 25 registered & 18 unregistered users

ROC – HMM vs Decision Trees (DT)



- DT is from: Y. Sheng, V. V. Phoha and S.M. Rovnyak, A Parallel Decision Tree-Based Method for User Authentication Based on Keystroke Patterns. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 2005.

HMM: AUTHENTICATION THROUGH KEYSTROKE DYNAMICS

Michele Rossi
rossi@dei.unipd.it

Dept. of Information Engineering
University of Padova, IT

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE