Information Leakage in Encrypted Network Traffic

Attacks and Countermeasures

Scott Coull
RedJack

Joint work with:

- Charles Wright (MIT LL)
- Lucas Ballard (Google)
- Fabian Monrose (UNC)
- Gerald Masson (JHU)

Encrypted Network Traffic

<u>Problem</u>: How do we stop eavesdropping?

Cryptography to the rescue!

Encrypted Network Traffic

- Encrypt the payload contents
- What about features of the network traffic itself?
  - Port numbers, packet sizes, timing info

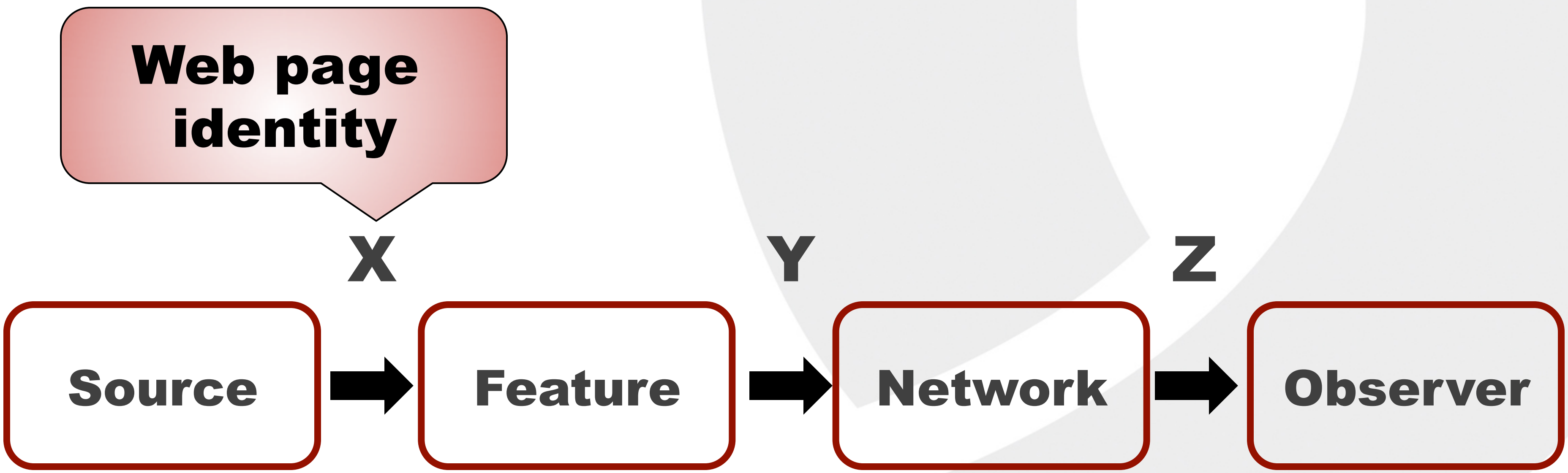
Network Traffic Features

- Traffic features may introduce a communications channel
  - Like a side-channel attack or covert channel

Source $\xrightarrow{X}$ Feature $\xrightarrow{Y}$ Network $\xrightarrow{Z}$ Observer

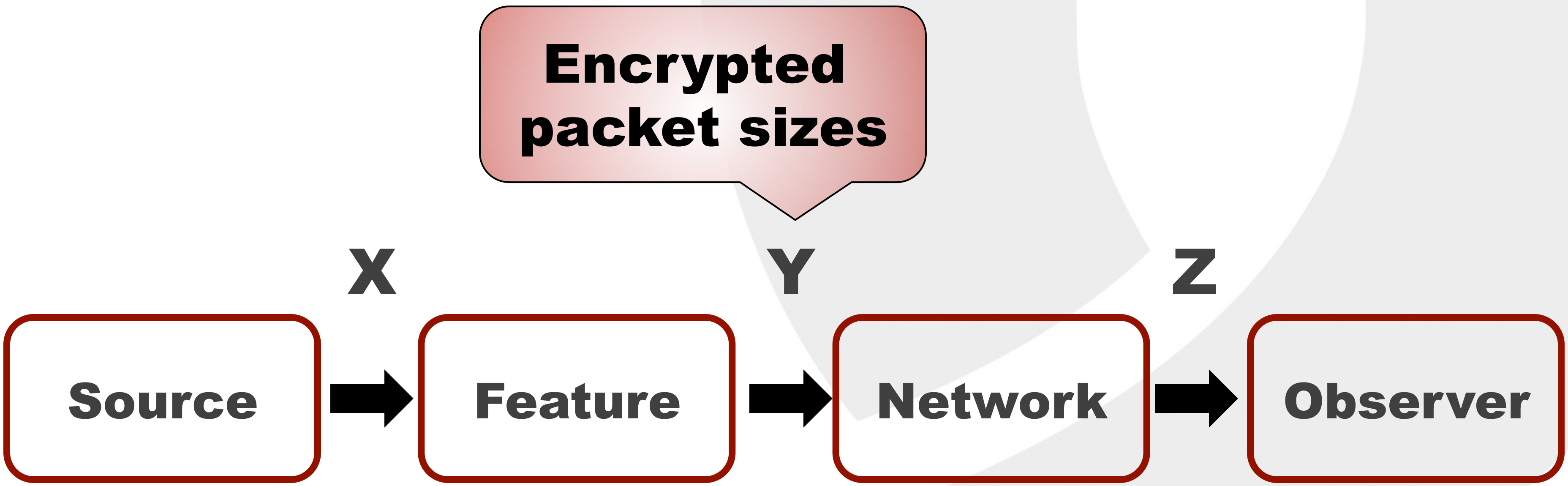
Network Traffic Features

- Traffic features may introduce a communications channel
  - Like a side-channel attack or covert channel



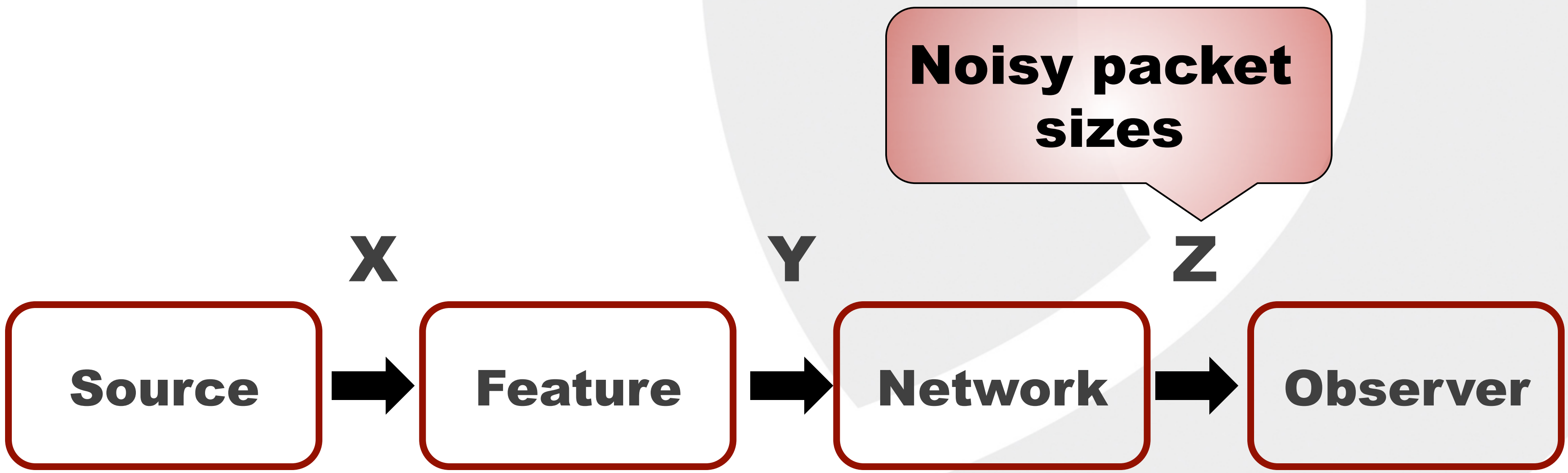
Network Traffic Features

- Traffic features may introduce a communications channel
  - Like a side-channel attack or covert channel



Network Traffic Features

- Traffic features may introduce an communications channel
  - Like a side-channel attack or covert channel



Preventing Information Leakage

- Information theory says we should make the distributions **uniform**
  - Quantize outputs to coarser levels
  - Ex: two vs. ten potential packet sizes
- What about performance?
  - Padding VoIP can introduce >42% overhead!!
  - Padding to 1500 bytes induces up to 150% overhead for HTTP!!

Implementation Problems

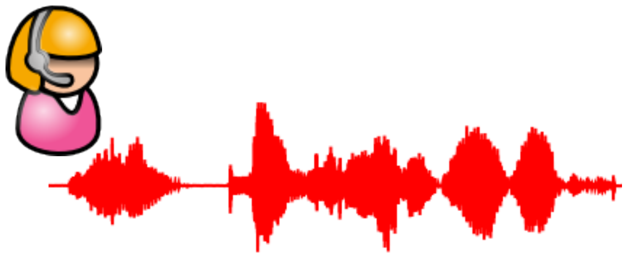
- Performance almost always wins over security for consumer implementations
- Results in attacks that infer contents of encrypted network traffic
  - Login passwords for SSH → packet timing
  - Web page identities for SSL/TLS → packet size
  - Language and phrases for VoIP → packet size

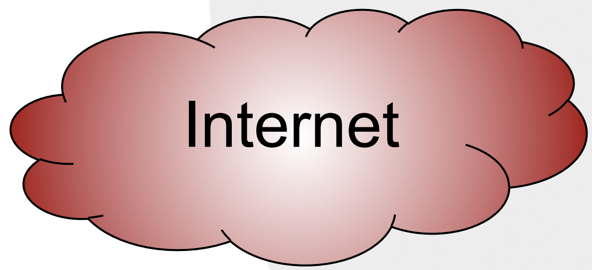
Overview

- Illustrate potential privacy problems introduced by network traffic features:
  - Uncovering Spoken Phrases in Encrypted Voice over IP Conversations
- Present efficient countermeasure by moving from information theoretic model:
  - Traffic Morphing: An Efficient Defense Against Statistical Traffic Analysis

How Does VoIP Work?

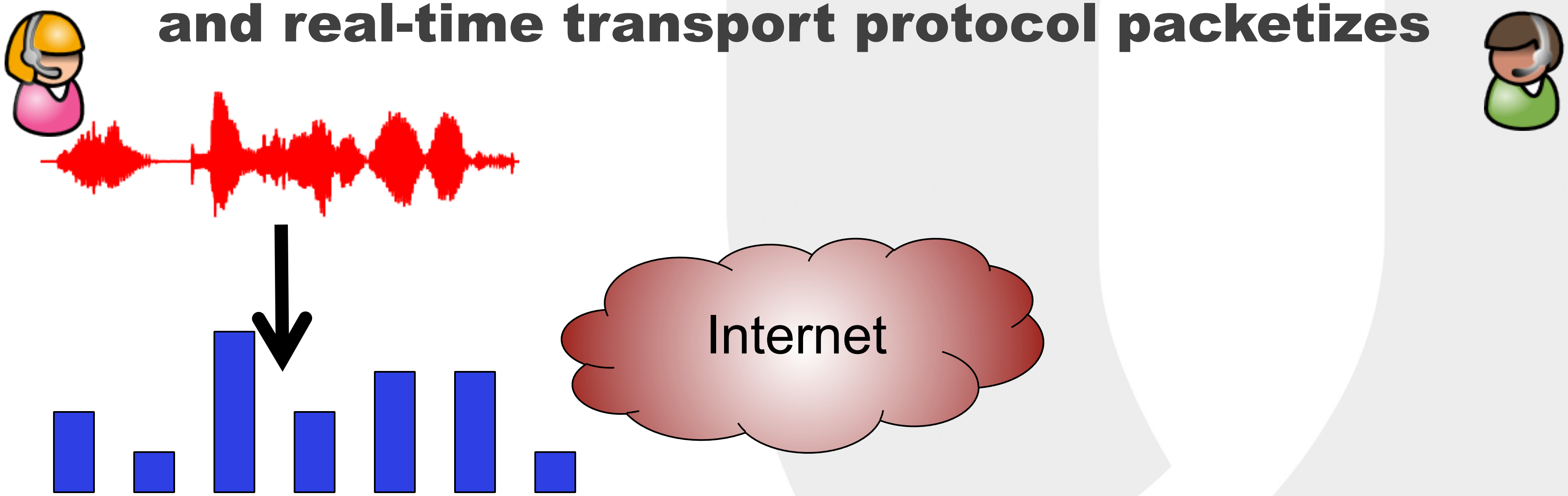
Alice speaks into her microphone to produce an audio waveform





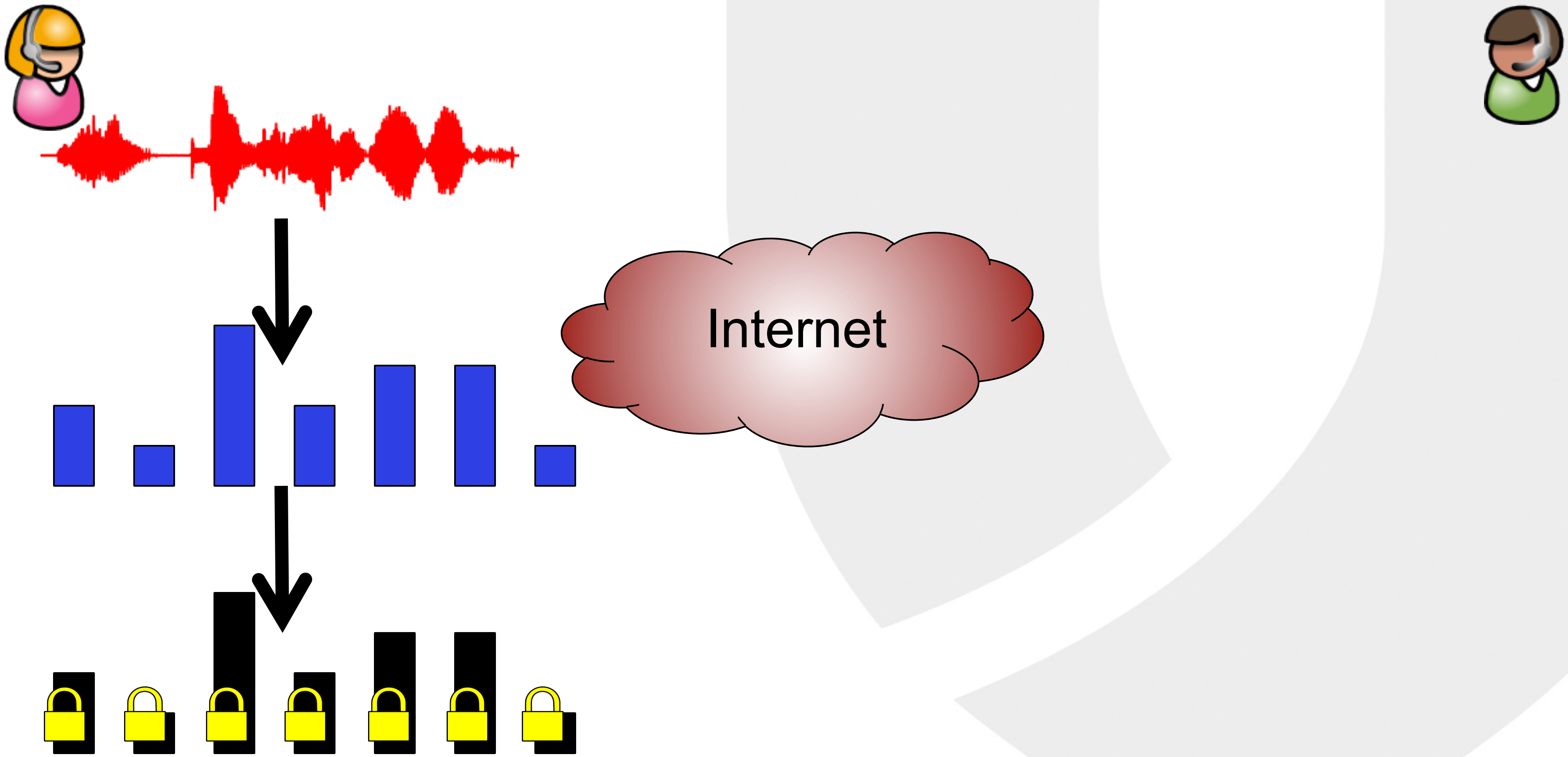
How Does VoIP Work?

Variable Bit-Rate audio codec compresses waveform and real-time transport protocol packetizes



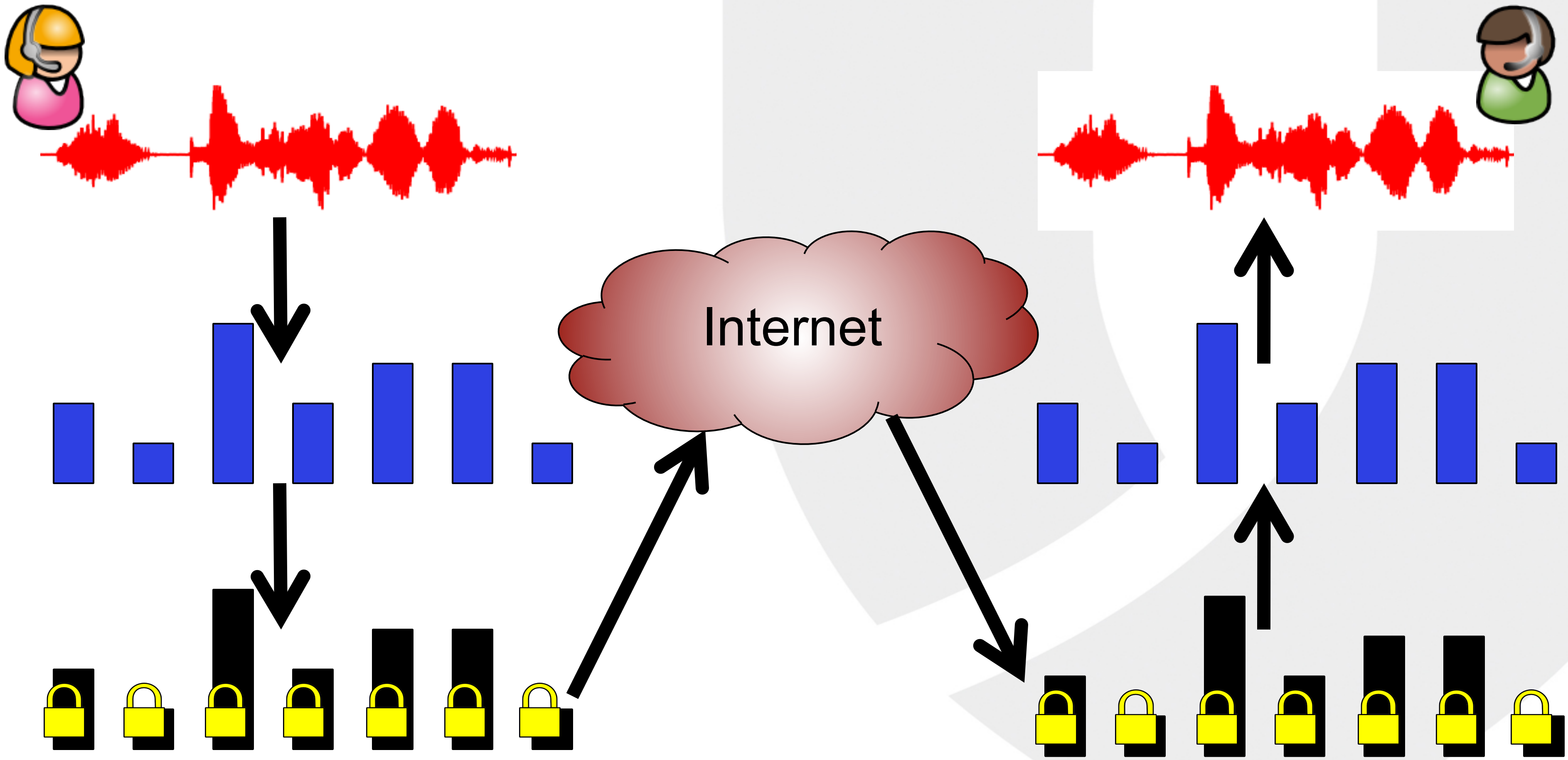
How Does VoIP Work?

Stream cipher used to efficiently encrypt packets



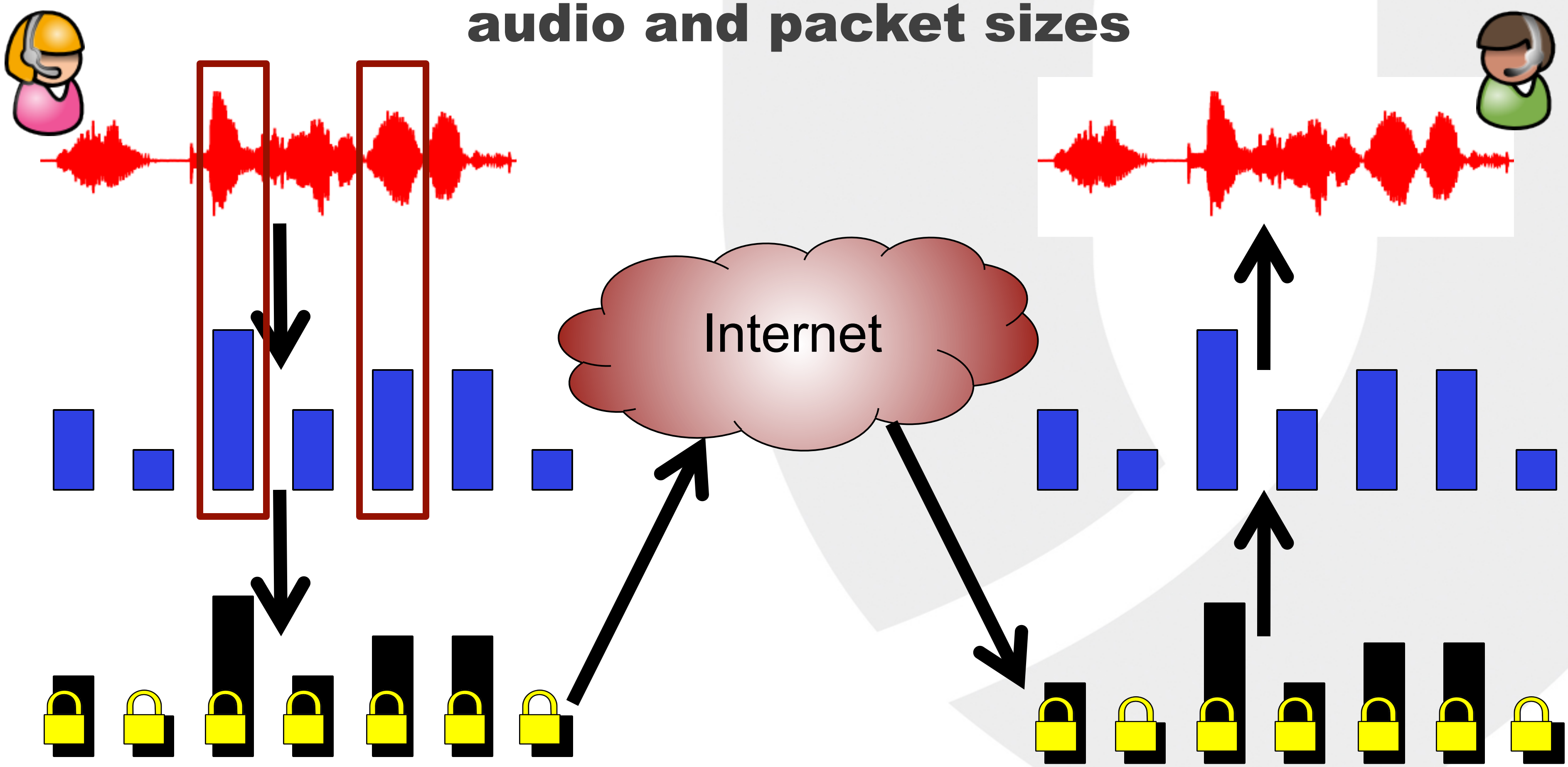
How Does VoIP Work?

Packets sent over IP network and decoded



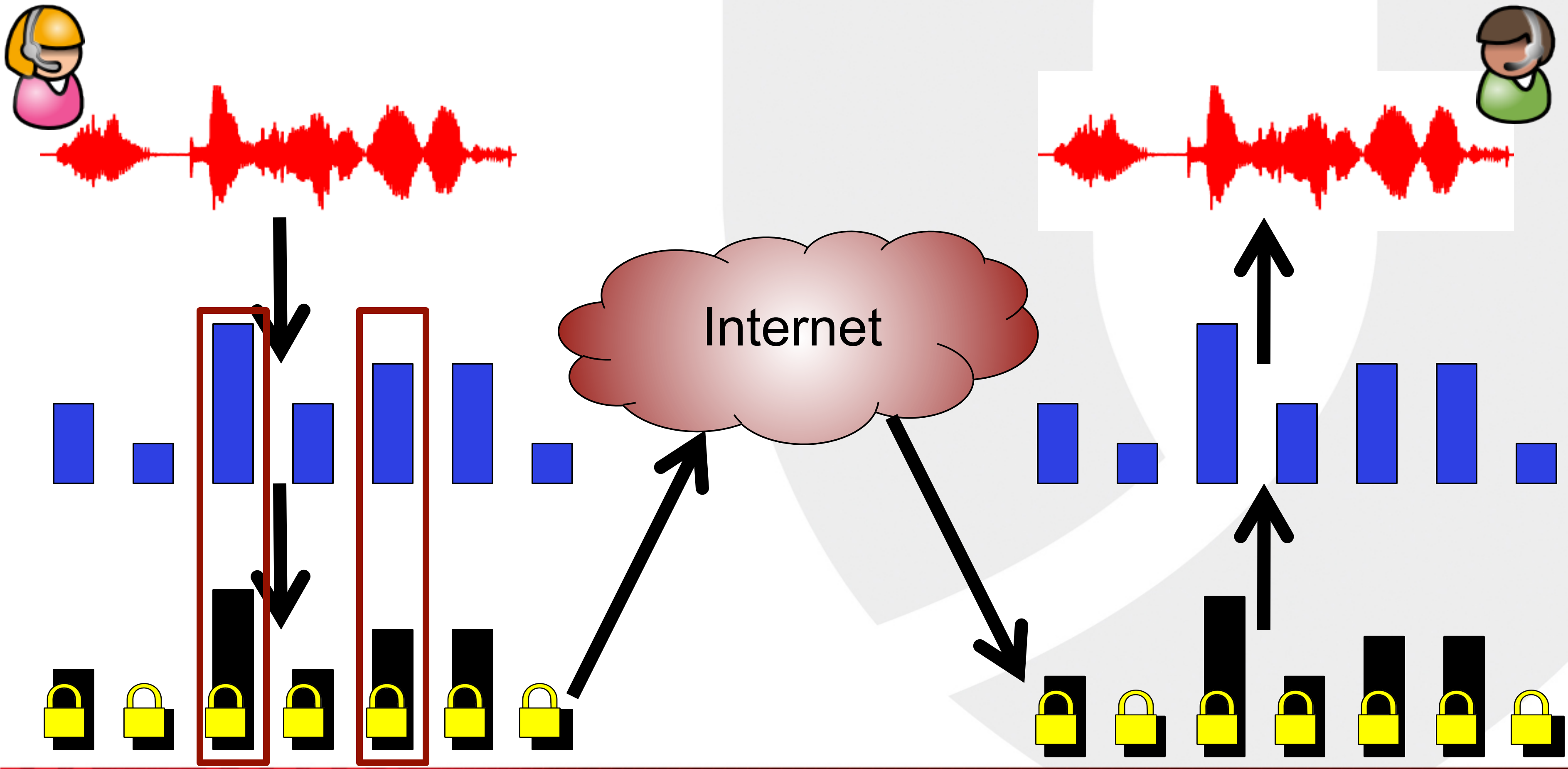
Attack Intuition

Variable Bit-Rate **codec allows correlation between audio and packet sizes**

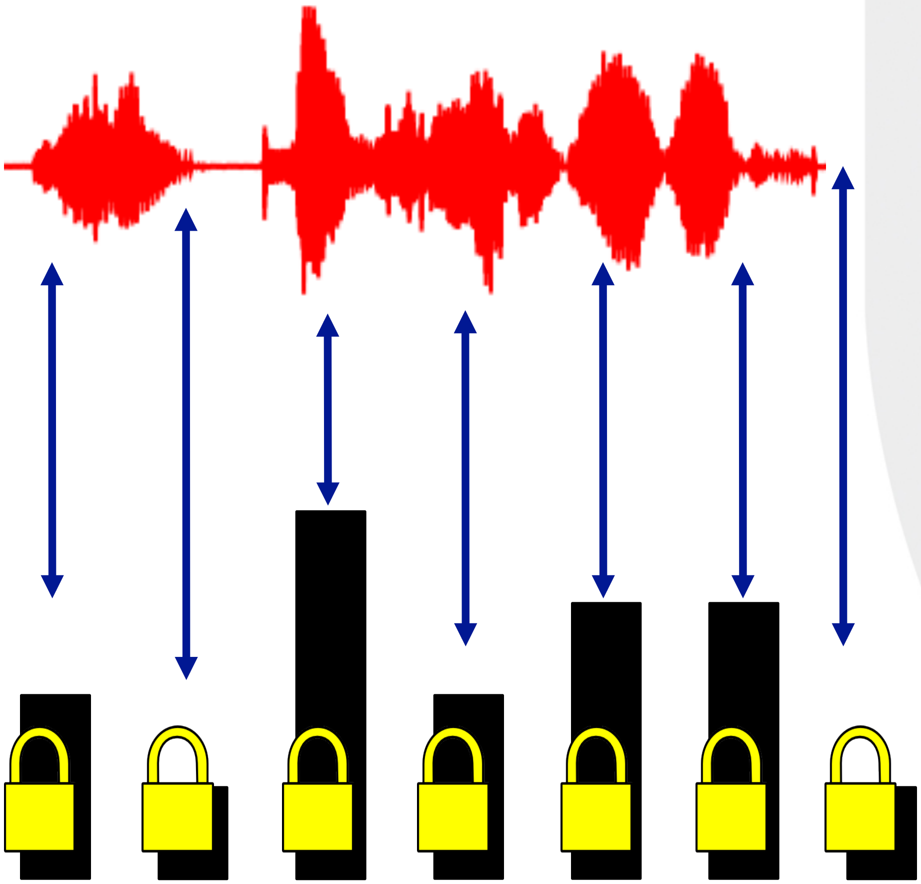


Attack Intuition

Length-preserving **stream cipher keeps correlation**



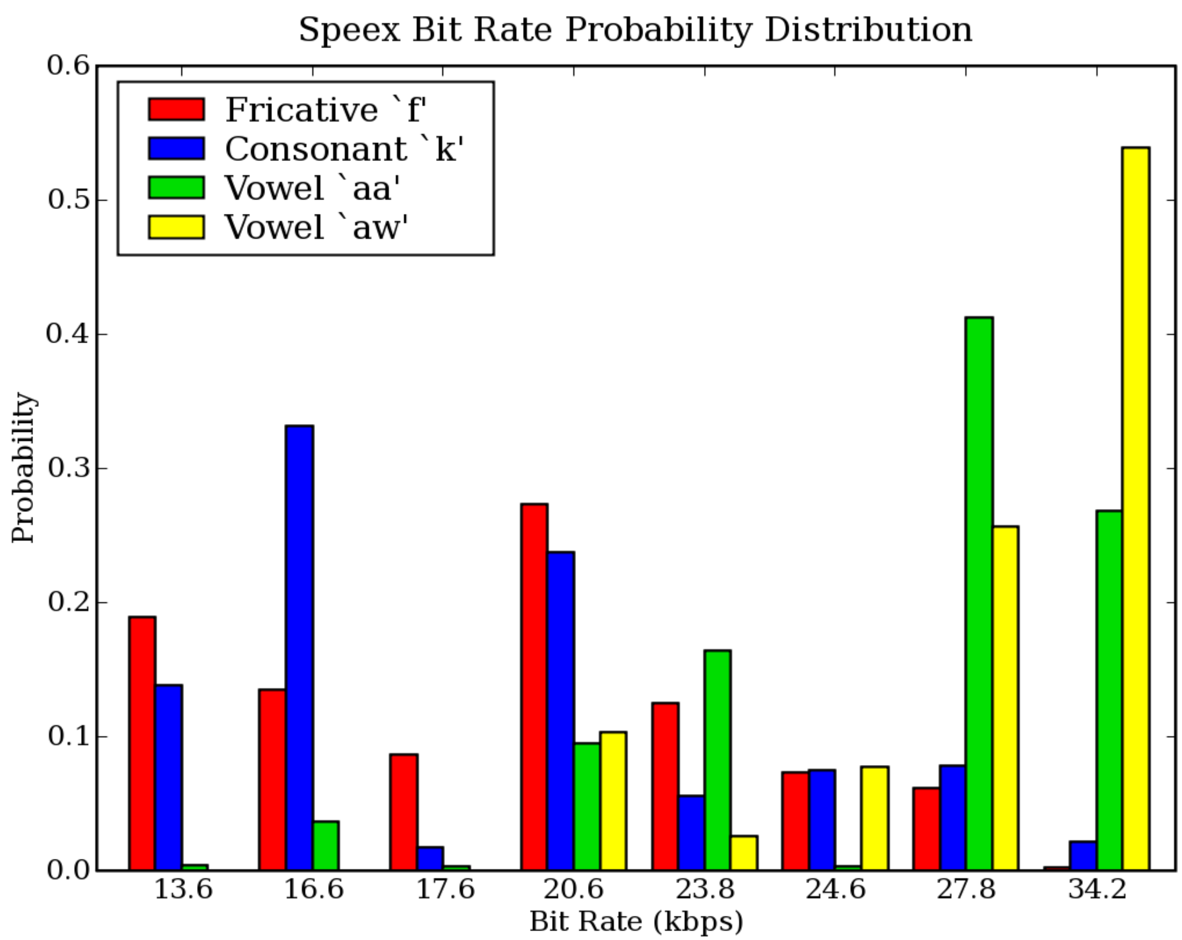
Attack Intuition



Packet sizes carry information about the original audio

Original audio is made up of the building blocks of language called phonemes

Attack Intuition



Phrase Spotting

- How do we use the correlation between packet size and audio?
  - Build a **statistical model** for the sequence of packet sizes in phrase
  - Use the model to detect when a phrase of interest has occurred

- Known as **phrase spotting**

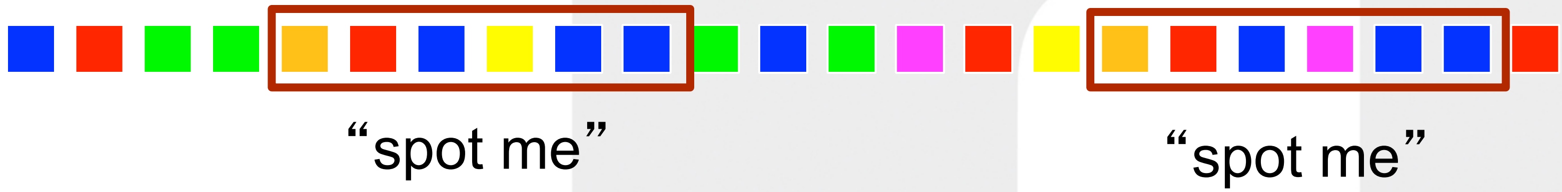
Phrase Spotting

Packets: 

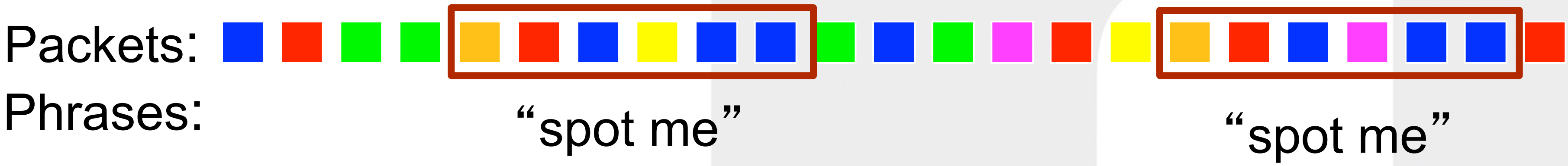
Phrase Spotting

Packets:

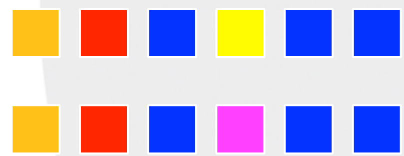
Phrases:



Phrase Spotting



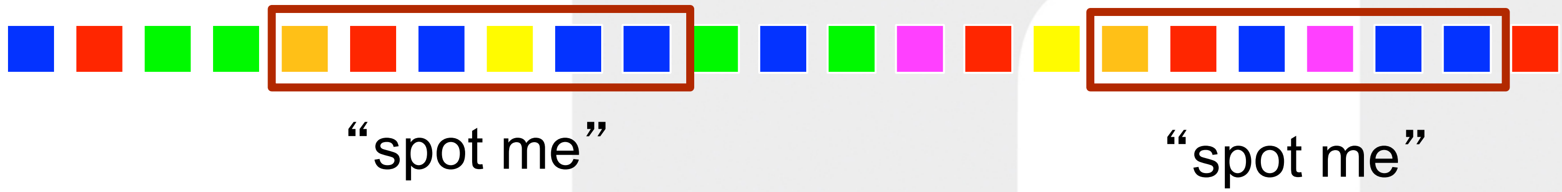
Extract Training Data



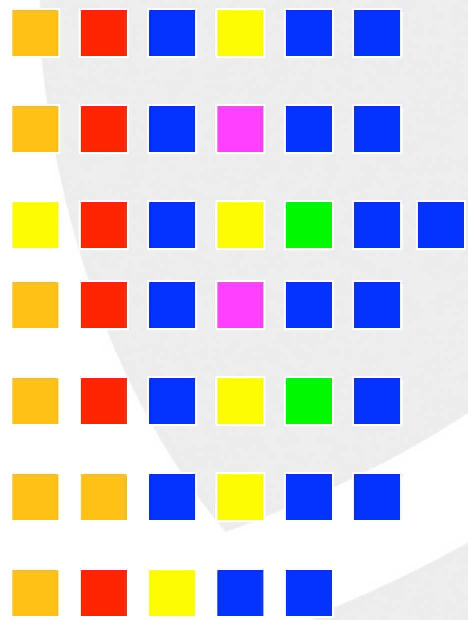
Phrase Spotting

Packets:

Phrases:



Extract Training Data

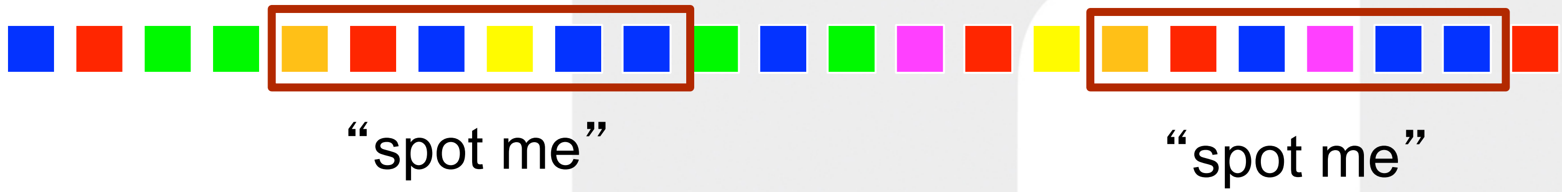


Phrase Spotting

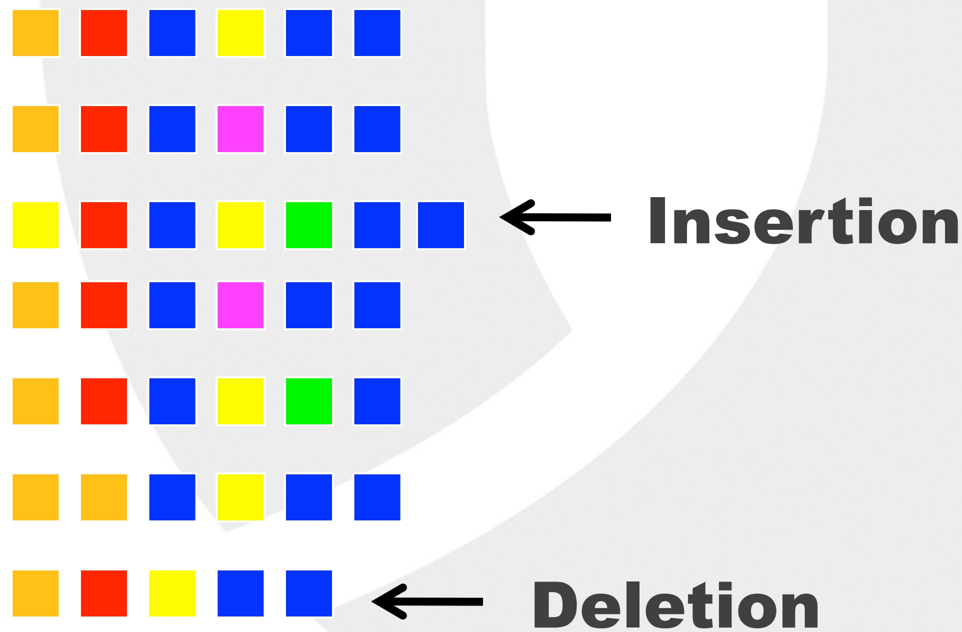
REDJACK

Packets:

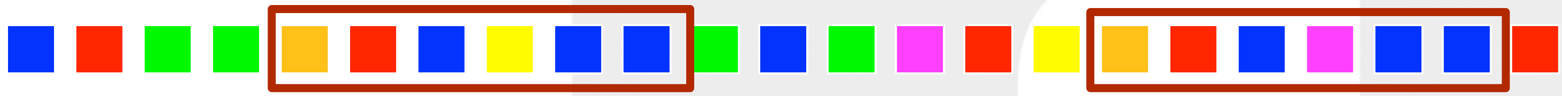
Phrases:



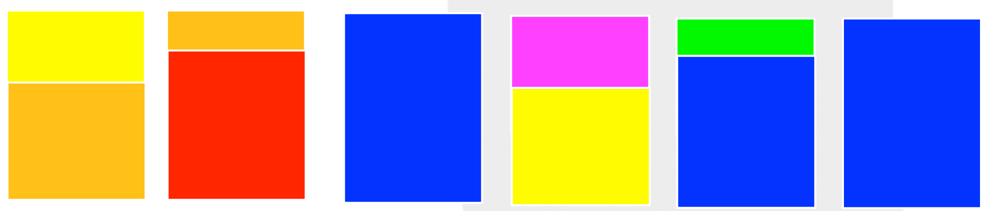
Multiple Sequence Alignment



Phrase Spotting

Packets: 

Phrases: “spot me” “spot me”



Aligned columns known as the consensus sequence for the phrase

Represented with a Profile Hidden Markov Model (HMM)

Synthesizing Training Data

- How do we train our phrase HMM without speaker-specific data?
- What if we do not have examples of the phrase we are looking for?
- Use **concatenative synthesis**!
  - Break phrase pronunciation into phonemes
  - Concatenate available phonemes together

Synthesizing Training Data

1. Split phrase into words

the bike is red

Synthesizing Training Data

1. Split phrase into words

   the bike is red

2. Break words into phonemes

   dh ah   b ay k   ih z   r eh d

Synthesizing Training Data

1. Split phrase into words

   the bike is red

2. Break words into phonemes

   dh ah   b ay k   ih z   r eh d

3. Replace phonemes with packet sizes

Synthesizing Training Data

1. Split phrase into words

   the bike is red

2. Break words into phonemes

   dh ah   b ay k   ih z   r eh d

3. Replace phonemes with packet sizes

Synthesizing Training Data

1. Split phrase into words

   the bike is red

2. Break words into phonemes

   dh ah   b ay k   ih z   r eh d

3. Replace phonemes with packet sizes

REDJACK

Synthesizing Training Data

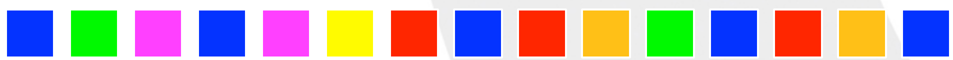
1. Split phrase into words

   the bike is red

2. Break words into phonemes

   dh ah   b ay k   ih z   r eh d

3. Replace phonemes with packet sizes



Synthesizing Training Data

1. Split phrase into words

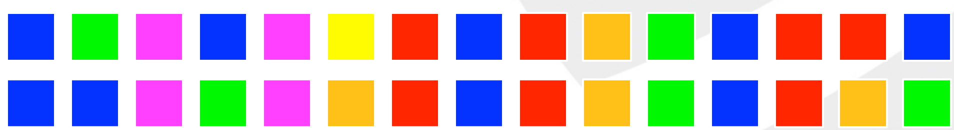
   the bike is red

2. Break words into phonemes

   dh ah   b ay k   ih z   r eh d

3. Replace phonemes with packet sizes

4. Repeat to capture variations in speech

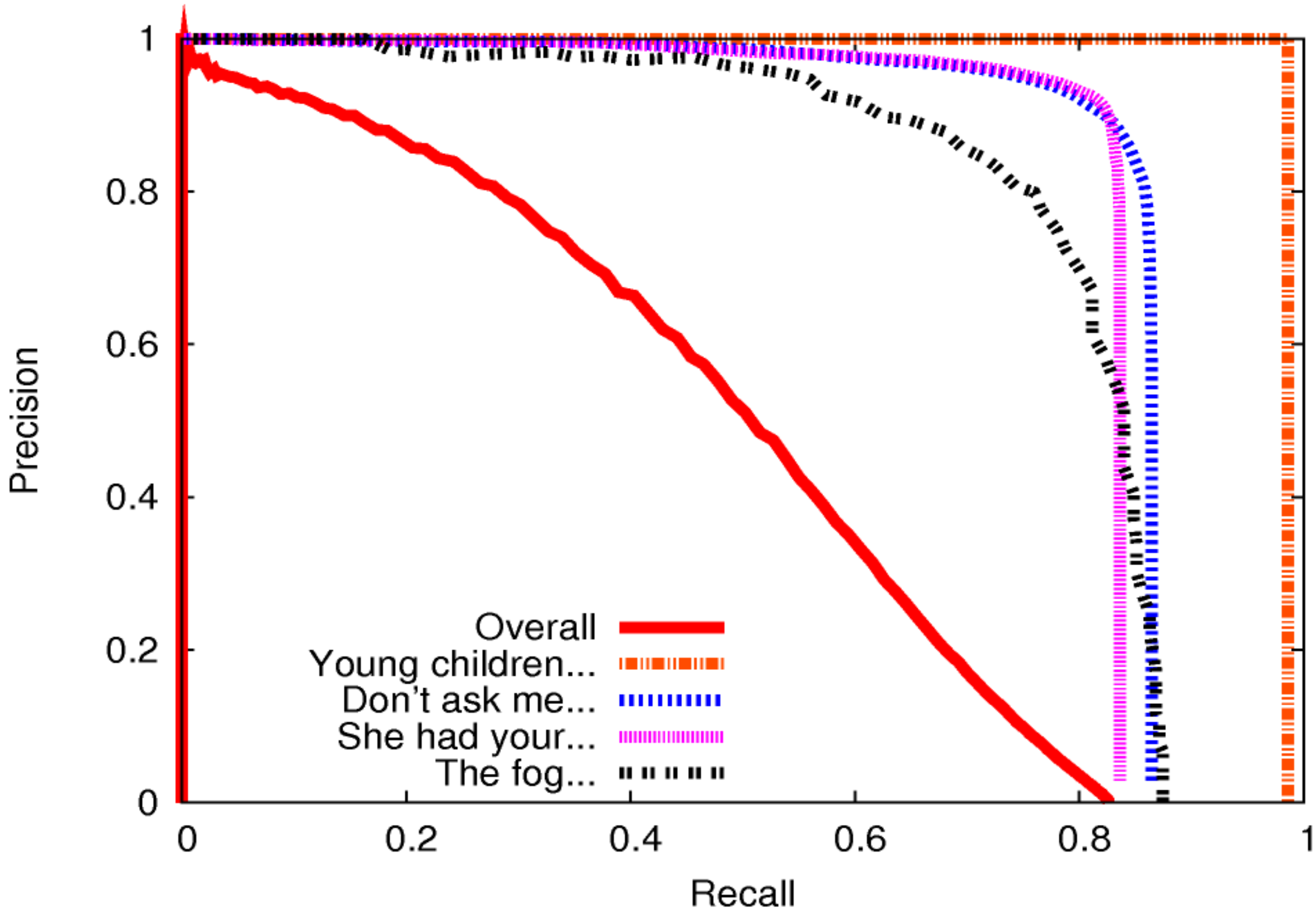


…

Evaluation

- **TIMIT Dataset:**
  - 462 training speakers, 168 testing speakers
  - 122 distinct phrases
  - Diverse set of dialects, pronunciations, etc.
- **Use information retrieval metrics:**
  - Recall = TP / (TP+FN)
  - Precision = TP / (TP+FP)

Results



Precision
Recall
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
Overall
Young children...
Don't ask me...
She had your...
The fog...

Results

- Method is **robust** to variations in the audio
  - Noise
  - Gender
  - Dialect
  - Wide-band vs. Narrow-band
- Surprising amount of information leakage from encrypted VoIP traffic
  - More recent work produces transcripts

Countermeasures

- Padding can be effective for some types of information, but not others
- Can we do better?

Protocol	Leaked Info.	Block Size	Accuracy	Overhead
VoIP	Phrases	256-bit	0.4%	16.5%
	Language	512-bit	6.9%	42.2%
HTTP	Web Pages	1500 bytes	35.9%	140.8%

Traffic Morphing

- What if we could **morph** one class of traffic to look like another?
  - Ex: Farsi appears to be English
- Use **convex optimization** to stochastically change packet sizes to:
  - Minimize overhead
  - Ensure distribution “looks” like the target

How to Morph Traffic

Source Distribution

(Packet Sizes for Farsi)

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

Target Distribution

(Packet Sizes for English)

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix}$$

How to Morph Traffic

Find morphing matrix A such that:

$$\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ & & \dots & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

How to Morph Traffic

Treat each column as a probability distribution for morphing source packet size s_j to target

$$\begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{bmatrix} = \begin{bmatrix} a_{11} & \boxed{a_{12}} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & & \cdots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix}$$

$$a_{ij} = \Pr[Y=s_i \mid X=s_j]$$

How to Morph Traffic

- Underspecified system of equations:
  - n^2 unknowns and 2n equations
  - Infinitely many solutions
- Choose one that minimizes the cost:

$$
\begin{array}{lll}
\text{minimize} & f_0(A) & \\
\text{subject to} & \sum_{j=1}^{n} a_{ij} x_j = y_i, & \forall i \in [1, n] \\
 & \sum_{i=1}^{n} a_{ij} = 1, & \forall j \in [1, n] \\
 & a_{ij} \geq 0, & \forall i, j \in [1, n]
\end{array}
$$

How to Morph Traffic

- **Underspecified system of equations:**
  - n^2 unknowns and 2n equations
  - Infinitely many solutions
- **Choose one that minimizes the cost:**

Overhead

$$
\begin{array}{lll}
\text{minimize} & f_0(A) & \\
\text{subject to} & \sum_{j=1}^{n} a_{ij} x_j = y_i, & \forall i \in [1, n] \\
& \sum_{i=1}^{n} a_{ij} = 1, & \forall j \in [1, n] \\
& a_{ij} \geq 0, & \forall i, j \in [1, n]
\end{array}
$$

How to Morph Traffic

- To morph source packet with size s_j:
  - Find j^{th} column of morphing matrix A
  - Sample from column according to distribution
  - Alter packet size to match sampled size s_i

$$\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ & & \dots & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

Real-world Challenges

- Many issues arise in the real-world:
  - Reducing packet sizes?
  - Large sample spaces?
  - Slight changes in distribution?
  - Not enough packets?

VoIP Morphing Results

- Apply morphing to VoIP language identification work of Wright et al.
  - Nearest-neighbor classifier
  - n-gram distributions of packet sizes
  - Can the classifier distinguish morphed from real?

Strategy	Overhead	Bigram Accuracy
No Padding	0.0%	71%
512-bit Padding	42.2%	50%
Morphing	15.4%	54%

REDJACK

VoIP Morphing Results

- Morpher must be at least as **"powerful"** as the classifier to reduce accuracy
  - Trigram classifier beats bigram morphing
  - ...but we can make a trigram morpher...

Strategy	Bigram Accuracy	Trigram Accuracy
No Padding	71%	76%
512-bit Padding	50%	50%
Morphing	54%	76%

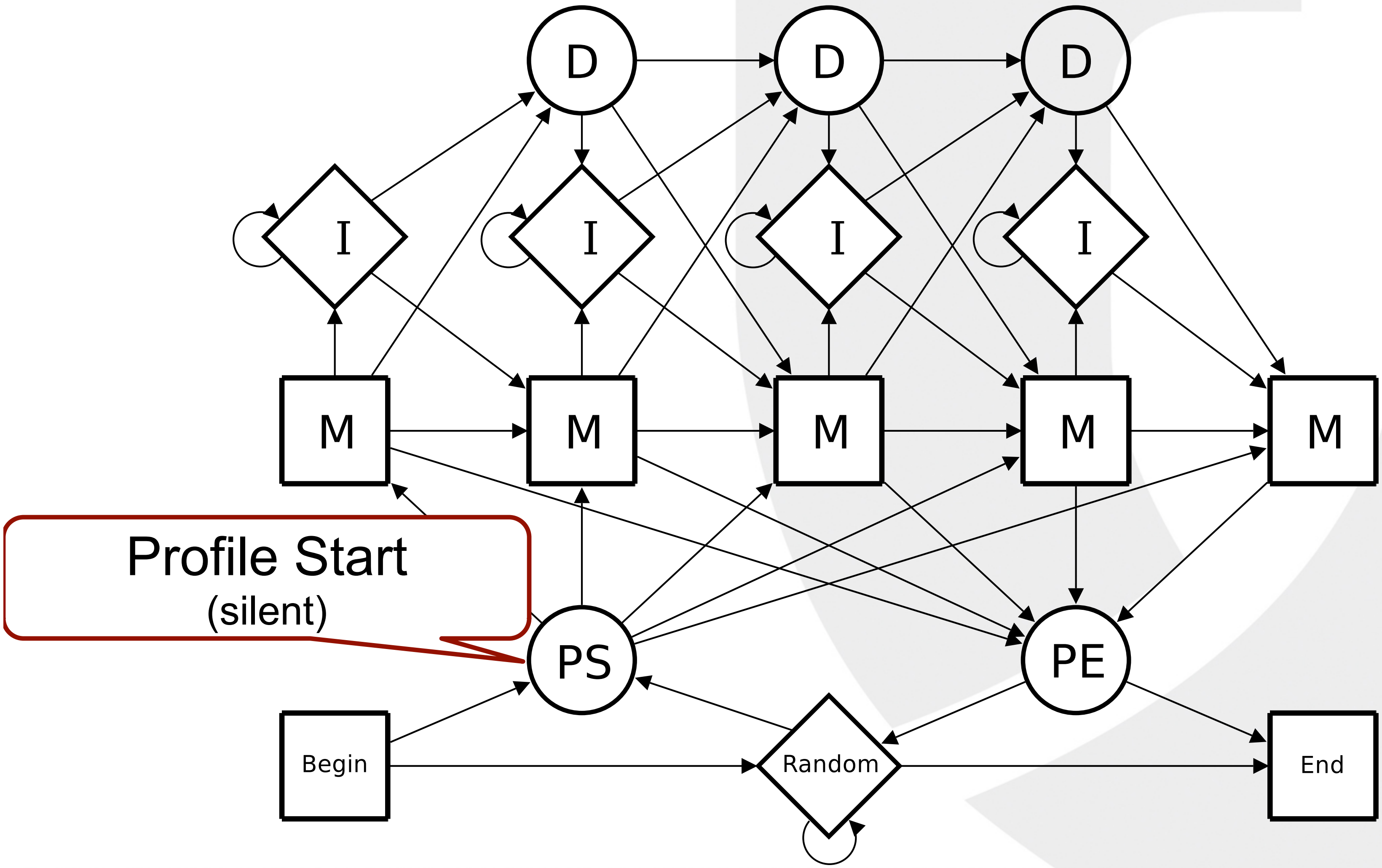
Open Questions

- How do we determine security of morphing-like countermeasures?
  - No obvious proof since we are not working in an information theoretic model
- Is there a way to tell if a classifier can ever win?
  - Certain natural limitations and costs may prevent morphing in some scenarios
  - Ex: Cannot buffer packets indefinitely

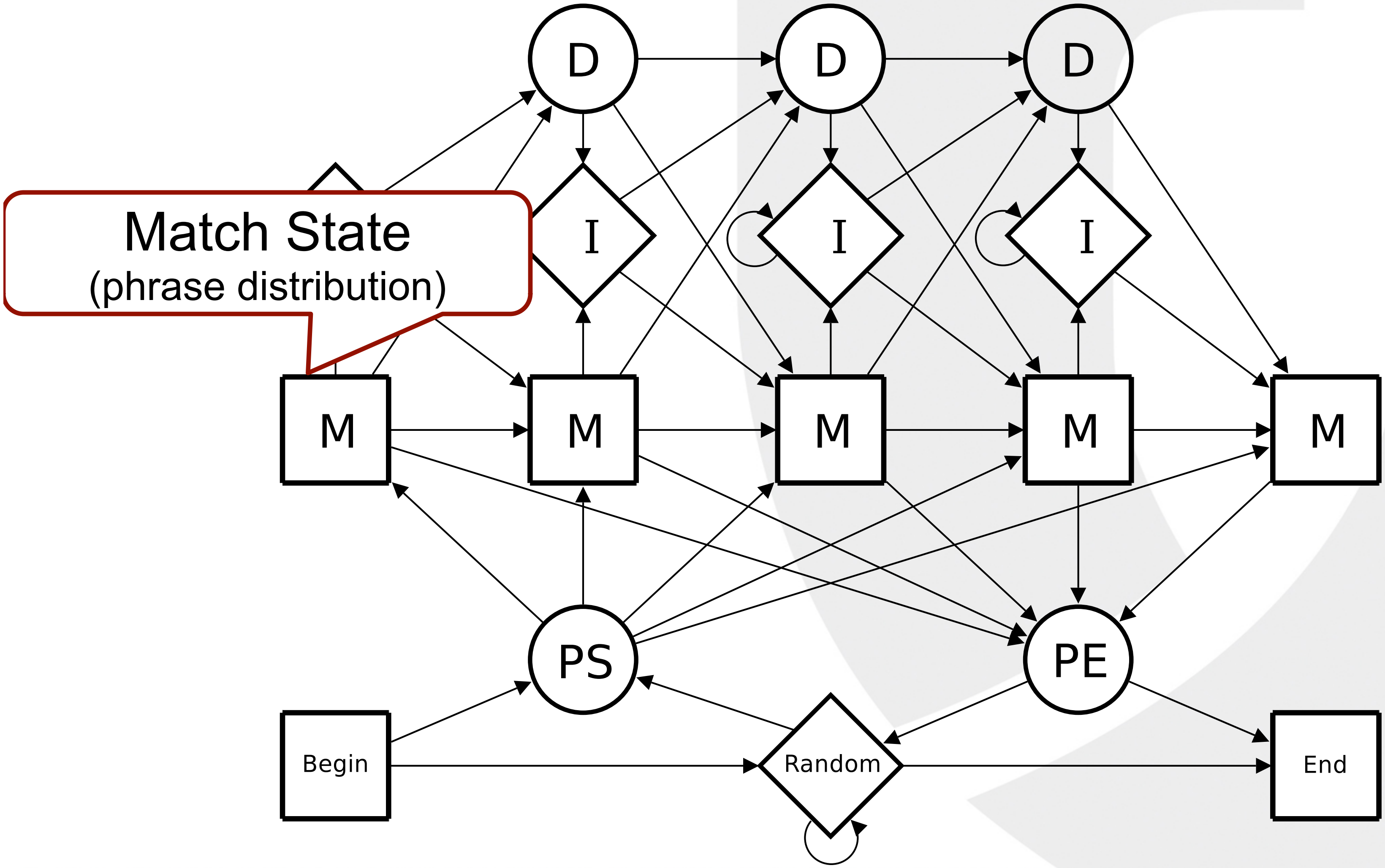
Summary

- Most users assume contents are secure with well-known cryptographic protocols
- Performance considerations lead to information leakage from traffic features
- Explored security/performance trade-off:
  - Phrase spotting in encrypted VoIP
  - Countermeasures using traffic morphing

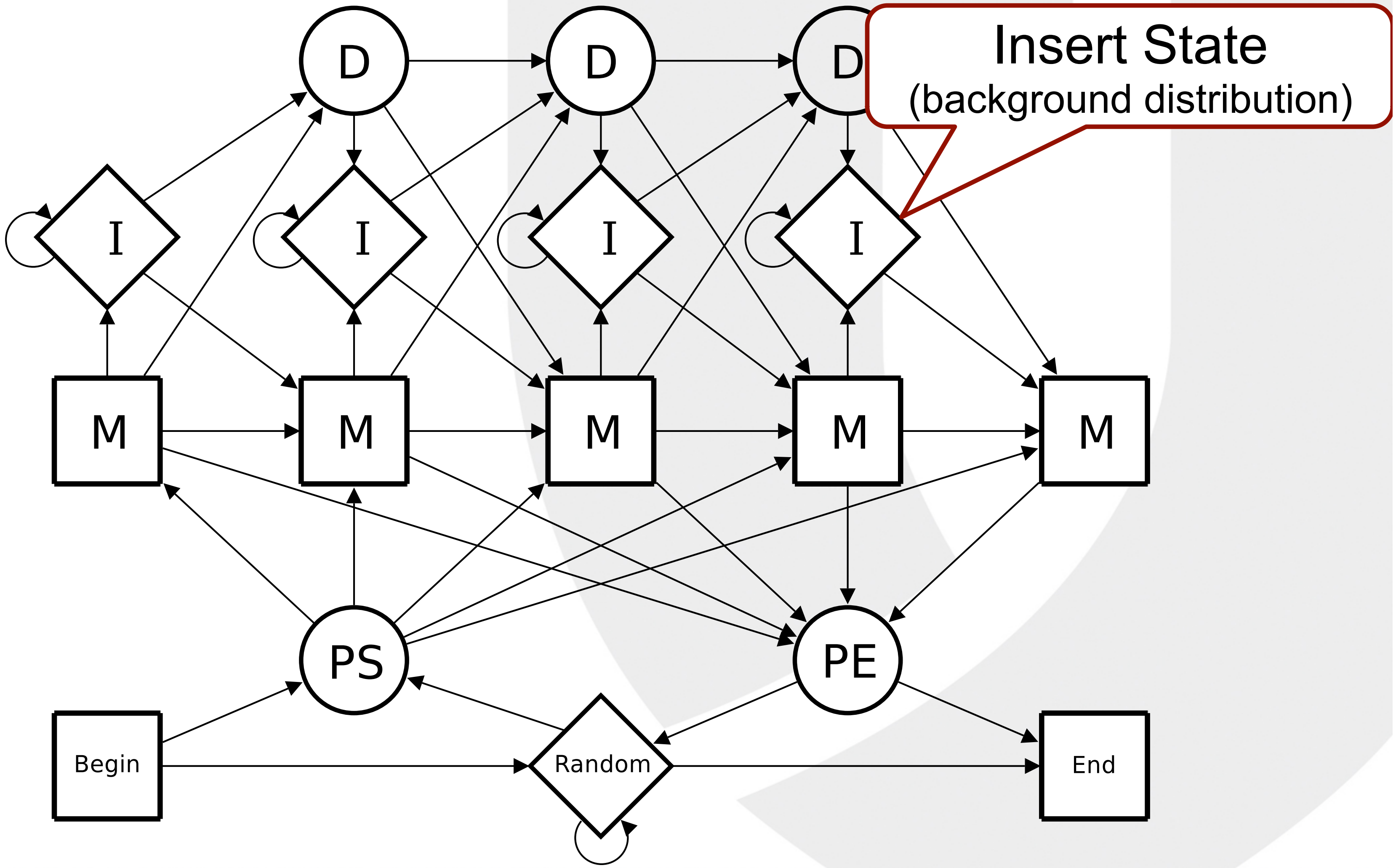
Search Hidden Markov Model



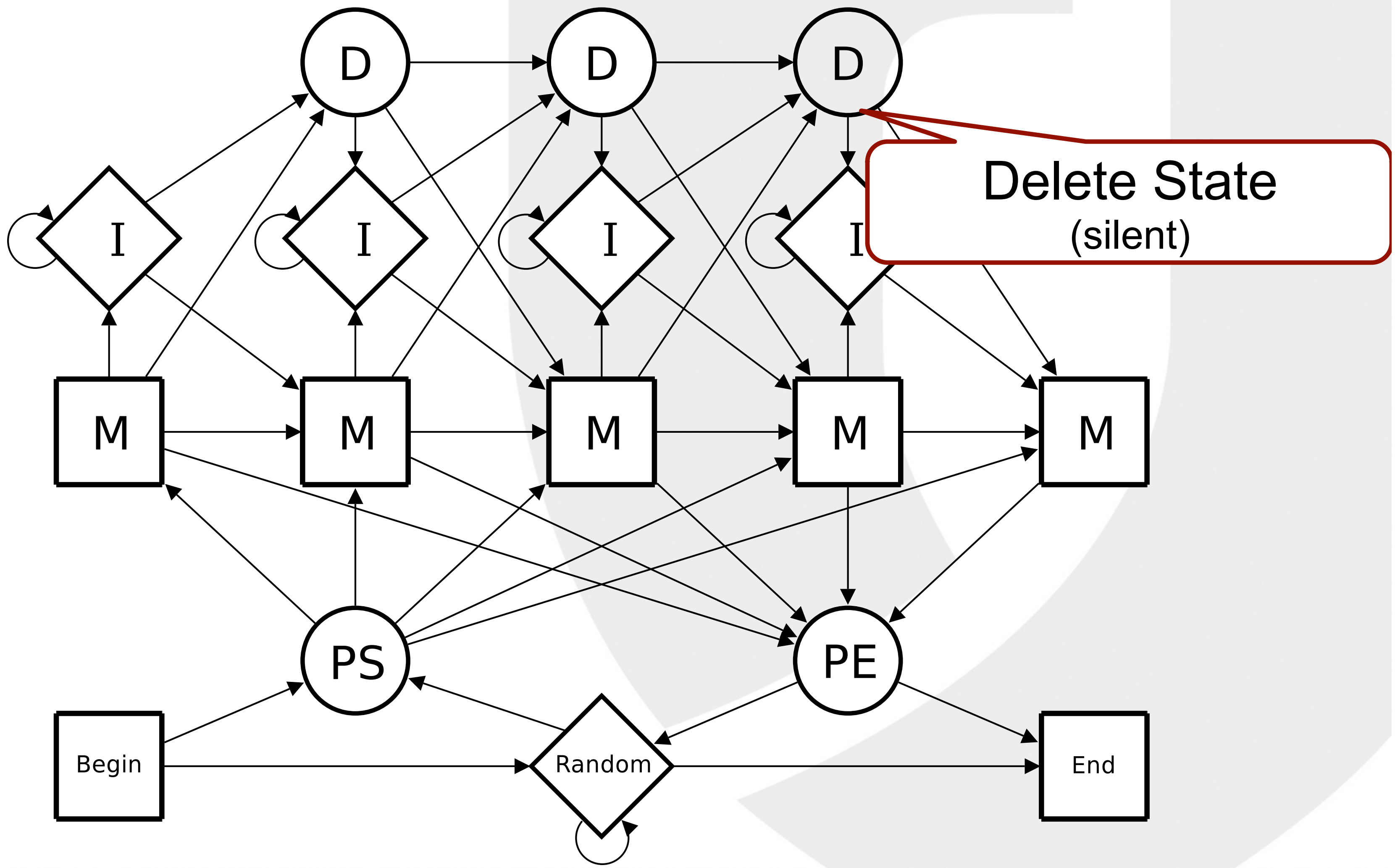
Search Hidden Markov Model



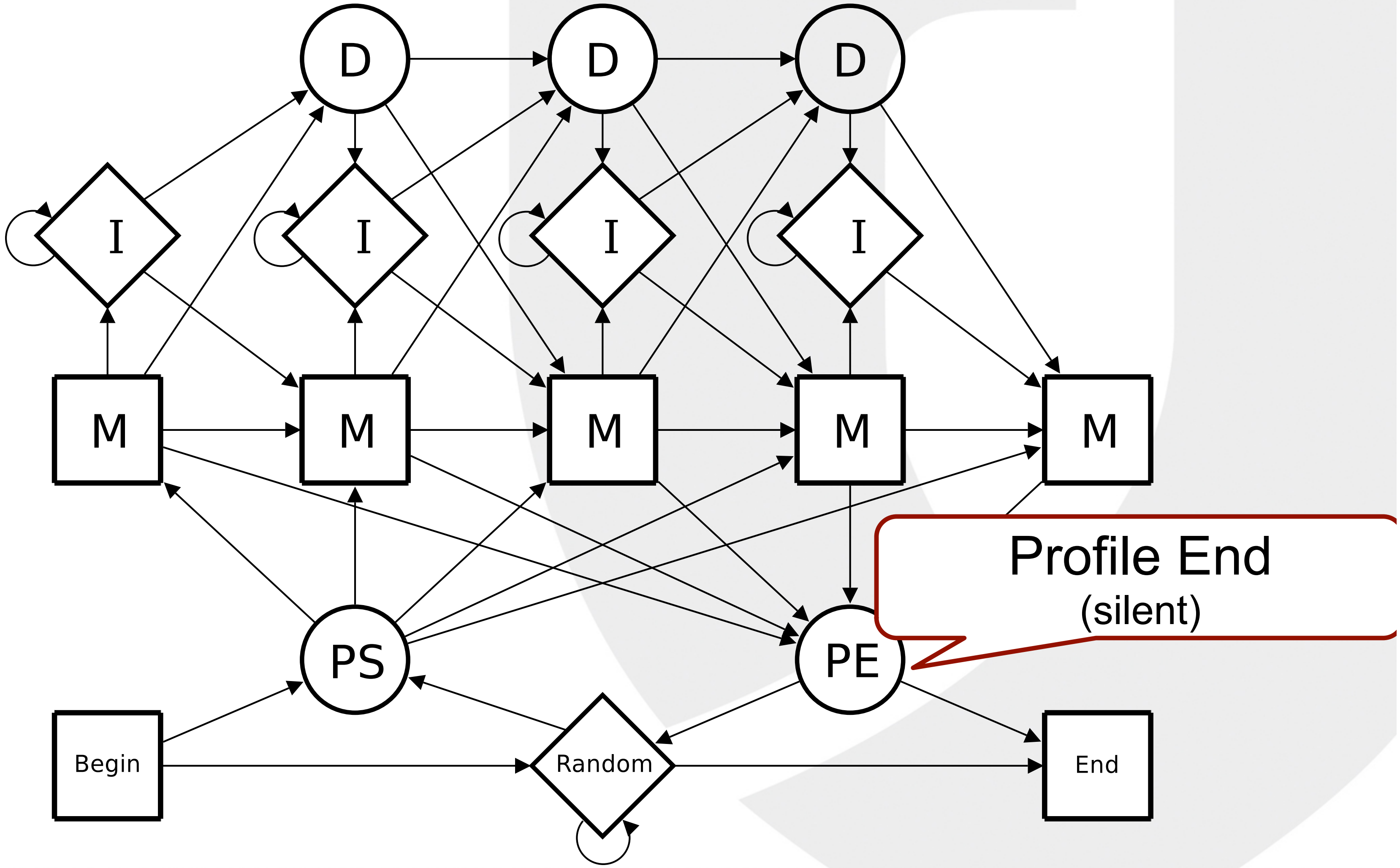
Search Hidden Markov Model



Search Hidden Markov Model



Search Hidden Markov Model



Search Hidden Markov Model