# Improved Cryptanalysis of Py

Paul Crowley

LShift Ltd

Royal Holloway Information Security Group Seminar, May 2006

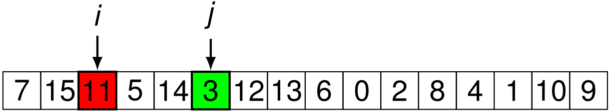
# Overview

- RC4
- Py
- SPP attack
- Our attack

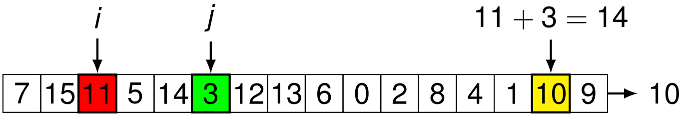
# RC4

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 15 | 11 | 5 | 14 | 3 | 12 | 13 | 6 | 0 | 2 | 8 | 4 | 1 | 10 | 9 |

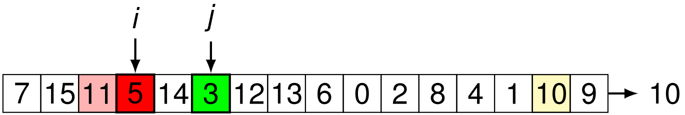
# RC4



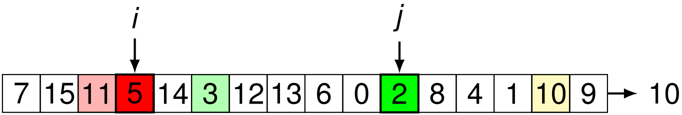
# RC4



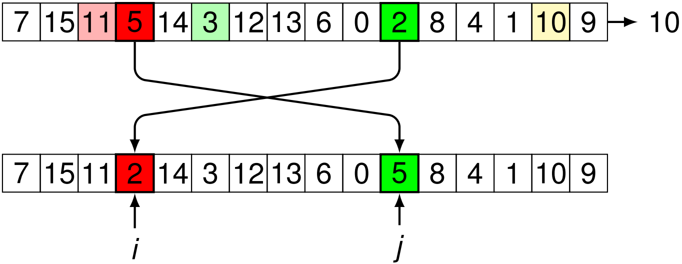
# RC4



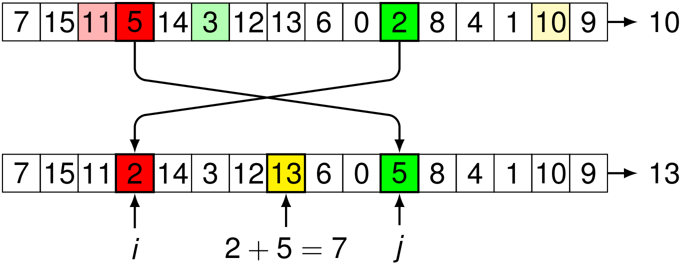
# RC4



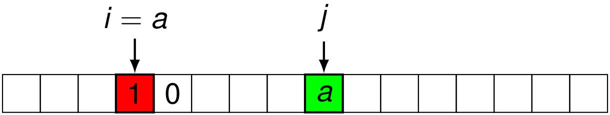
# RC4



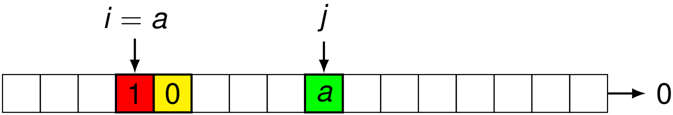
# RC4


7 15 11 5 14 3 12 13 6 0 2 8 4 1 10 9 → 10
7 15 11 2 14 3 12 13 6 0 5 8 4 1 10 9 → 13
i
2 + 5 = 7
j

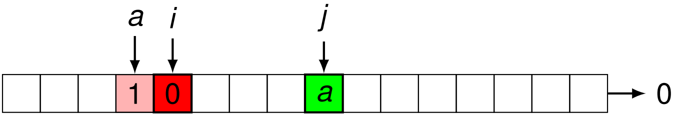
# Events in RC4



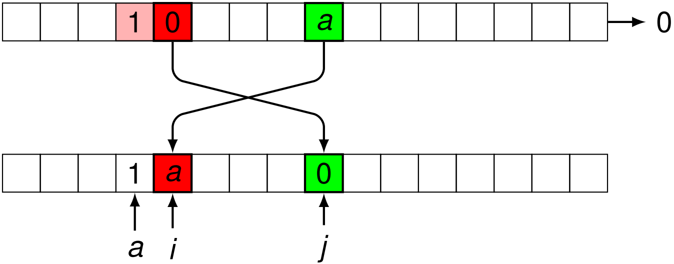
# Events in RC4



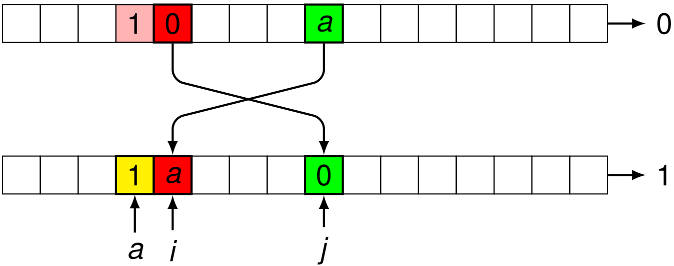
# Events in RC4



# Events in RC4



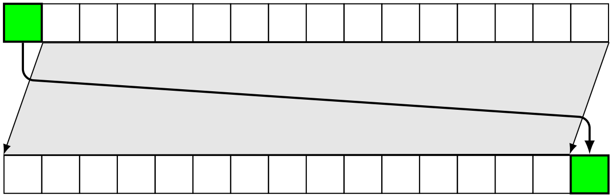
# Events in RC4



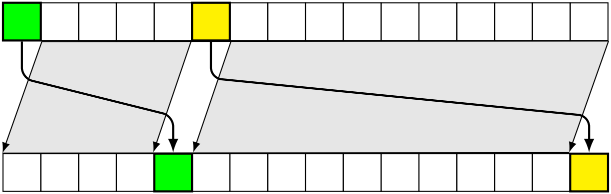
# Py

- eSTREAM entrant by Eli Biham and Jennifer Seberry
- Fast in software (2.6 cycles/byte on some platforms)
- SPP attack: $2^{88}$ bytes of output
- Our attack: $2^{72}$ bytes

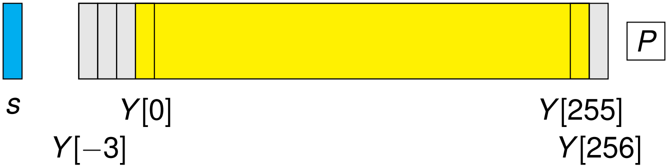
# Rolling arrays



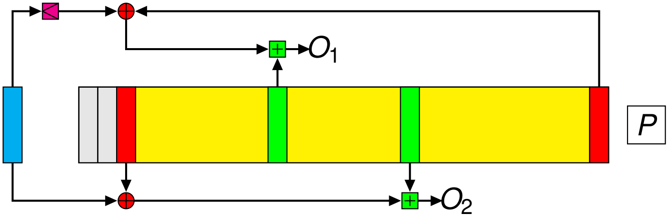
# Rolling swaps



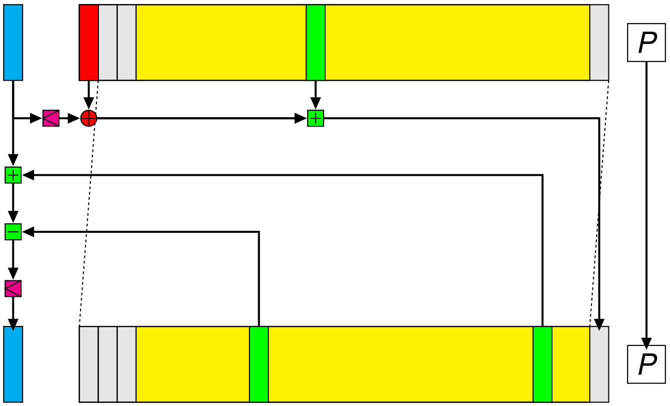
# Py internal state



# Output



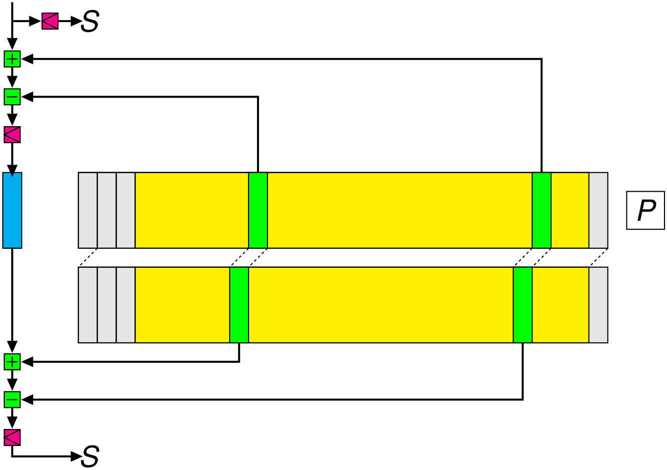
# Update



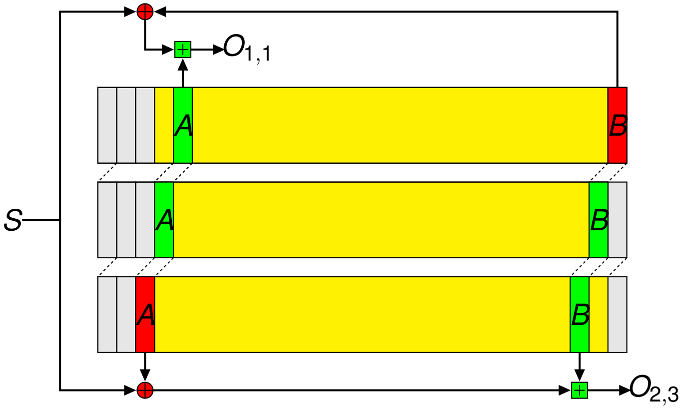
# SPP attack

- Gautham Sekar, Souradyuti Paul, Bart Preneel
- Defines event $L$ with $\Pr[L] \approx 2^{-41.91}$
- When $L$ occurs, two output bits are the same

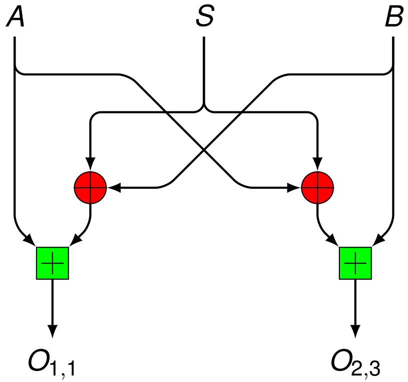
# Event $L$ (1)



# Event $L$ (2)



# Result of event $L$



# SPP distinguisher

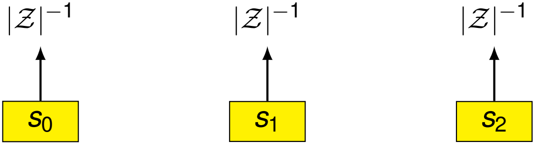
- Examine $2^{85}$ $O_{1,1}, O_{2,3}$ pairs (ie $2^{88}$ bytes)
- Count how many pairs have equal low bits
- Report “Py” if above a certain threshold, otherwise “random”
- How do we choose the threshold?

# Optimal distinguisher

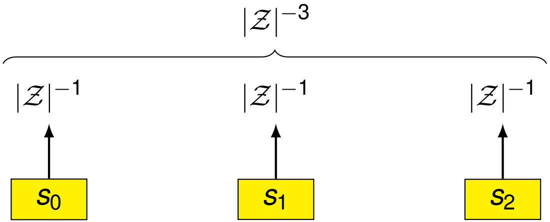
- Thomas Baignères, Pascal Junod, Serge Vaudenay
- Optimal distinguisher chooses the distribution which has the highest probability of producing the observed output
- Neyman-Pearson likelihood ratio test

# Optimal distinguisher

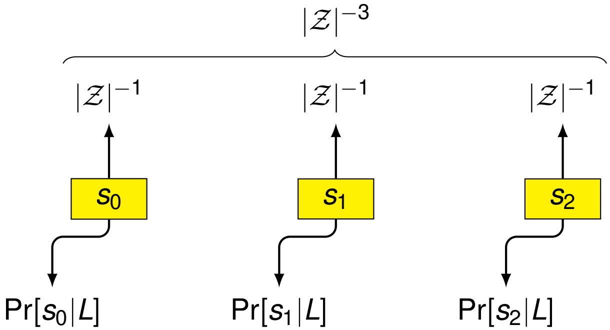
# Optimal distinguisher



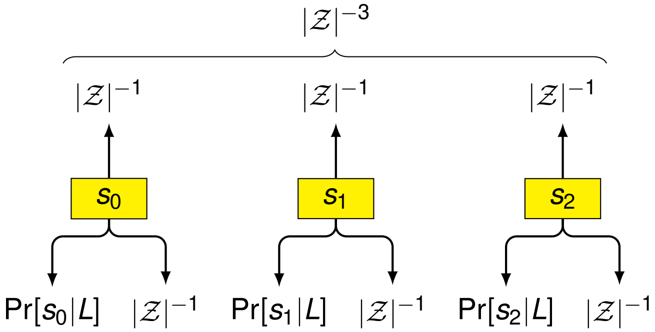
# Optimal distinguisher



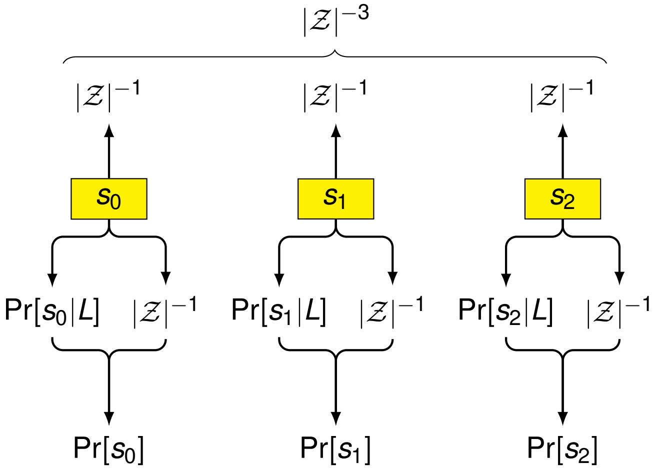
# Optimal distinguisher



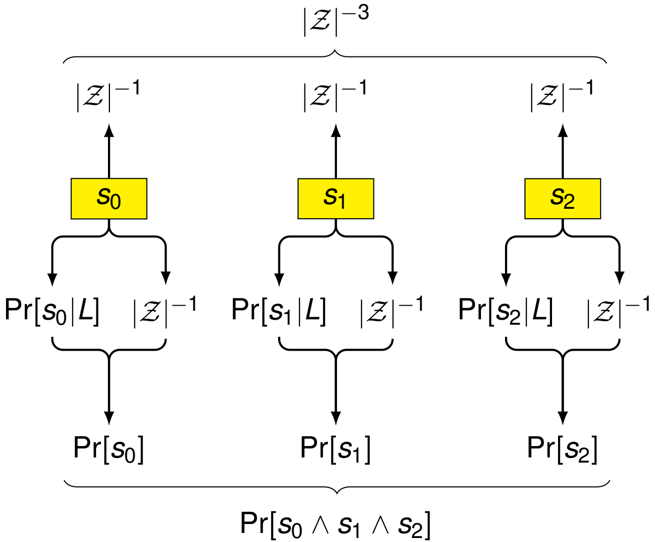
# Optimal distinguisher



# Optimal distinguisher



# Optimal distinguisher



# Optimal distinguisher

- We score each sample $s$ with $\mathrm{LLR}(s) = \log(\frac{\Pr[s|\text{Py}]}{\Pr[s|\text{Random}]})$
- Sum of scores is log-likelihood ratio for whole sample
- If score is positive, output Py
- Otherwise, output Random

# Efficacy of optimal distinguisher

- Where distribution is “close” to uniform random, efficacy $\beta = |\mathcal{Z}| \sum_{z \in \mathcal{Z}} \left( \Pr[z] - \frac{1}{|\mathcal{Z}|} \right)^2$

# Efficacy of optimal distinguisher

- Where distribution is "close" to uniform random, efficacy $\beta = |\mathcal{Z}| \sum_{z \in \mathcal{Z}} \left( \Pr[z] - \frac{1}{|\mathcal{Z}|} \right)^2$
- Need around $\frac{2}{\beta}$ samples for advantage $> \frac{1}{2}$

# Efficacy of optimal distinguisher

- Where distribution is "close" to uniform random, efficacy $\beta = |\mathcal{Z}| \sum_{z \in \mathcal{Z}} \left( \Pr[z] - \frac{1}{|\mathcal{Z}|} \right)^2$
- Need around $\frac{2}{\beta}$ samples for advantage $> \frac{1}{2}$
- If output only biased when event $L$ occurs: $\beta = \Pr[L]^2 \left( |\mathcal{Z}| \left( \sum_{z \in \mathcal{Z}} \Pr[z|L]^2 \right) - 1 \right)$

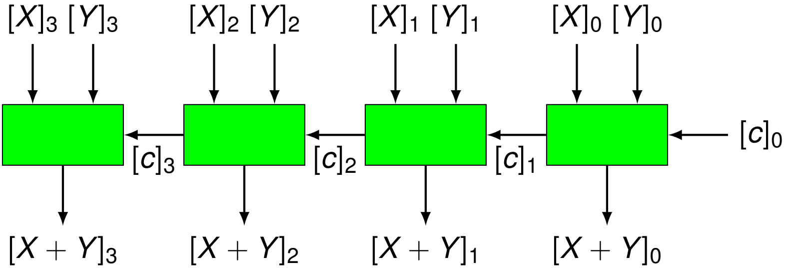
# Efficacy of optimal distinguisher

- Where distribution is "close" to uniform random, efficacy $\beta = |\mathcal{Z}| \sum_{z \in \mathcal{Z}} \left(\Pr[z] - \frac{1}{|\mathcal{Z}|}\right)^2$
- Need around $\frac{2}{\beta}$ samples for advantage $> \frac{1}{2}$
- If output only biased when event $L$ occurs: $\beta = \Pr[L]^2 \left(|\mathcal{Z}| \left(\sum_{z \in \mathcal{Z}} \Pr[z|L]^2\right) - 1\right)$
- SPP attack: $\beta = \Pr[L]^2$ so around $2^{85}$ samples

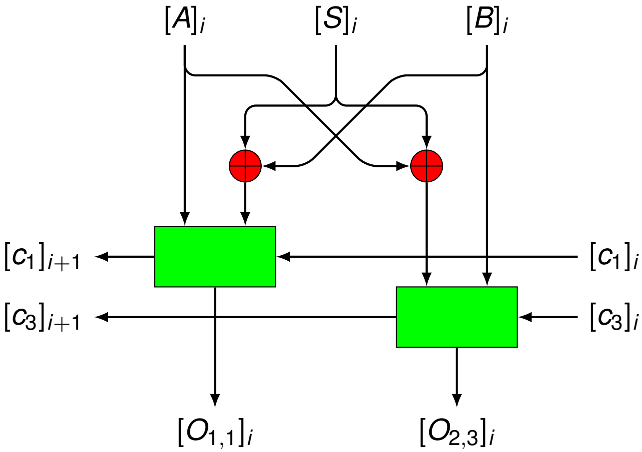
# Improving on the attack

- Use all bits of $O_{1,1}, O_{2,3}$
- Group output by column bitwise
- Find exact probability $\Pr[O_{1,1}, O_{2,3} = o_{1,1}, o_{2,3} | L]$
- Apply optimal distinguisher

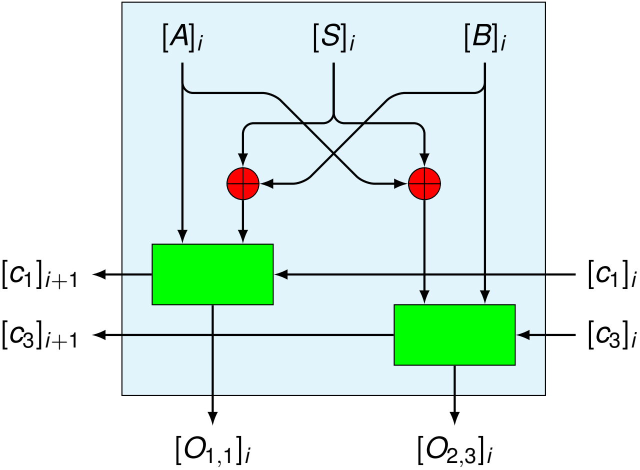
# Addition



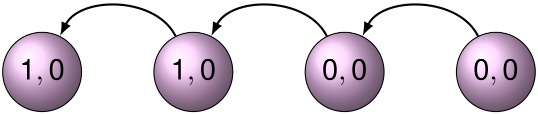
# Carry propagation



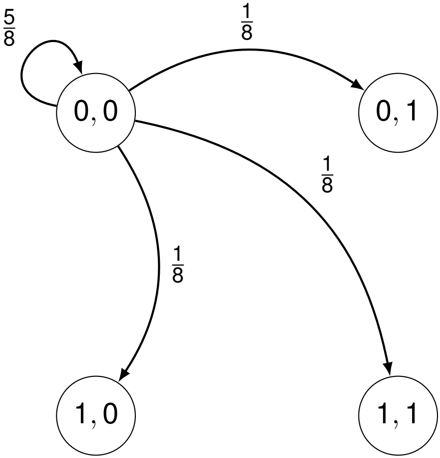
# Carry propagation



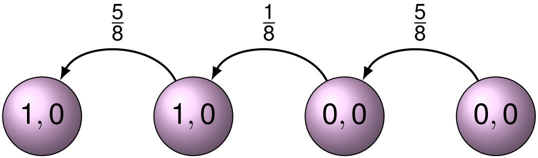
# Markov process



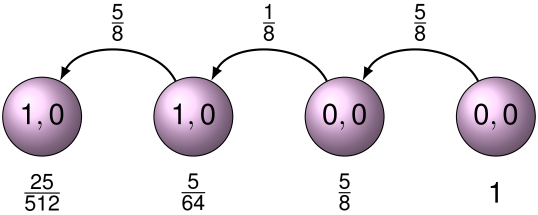
# Transition probabilities



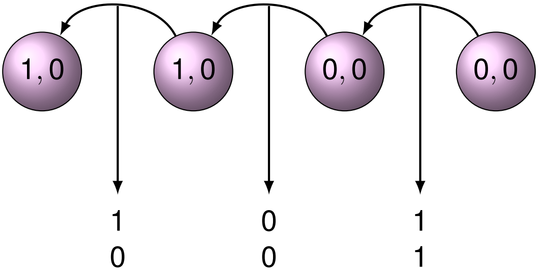
# Markov process



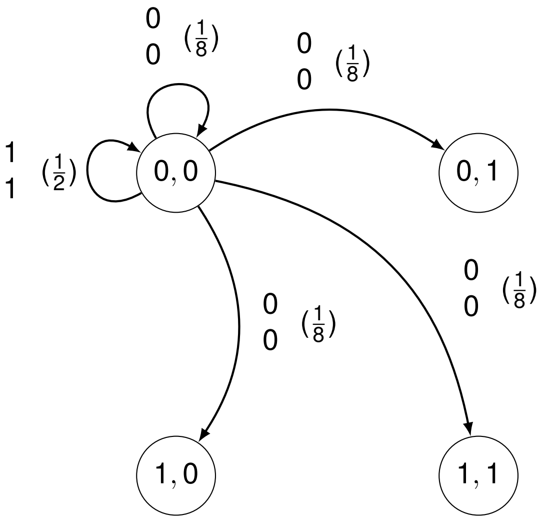
# Markov process



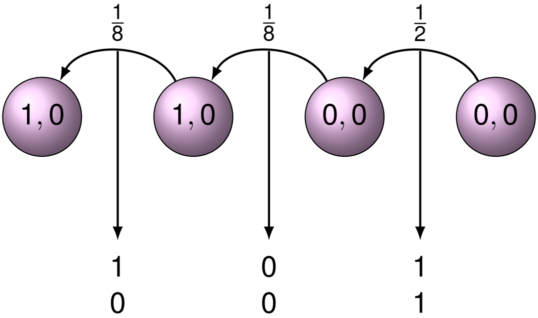
# Markov process



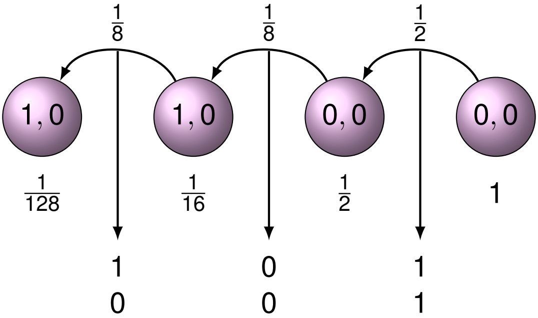
# Transition and output probabilities



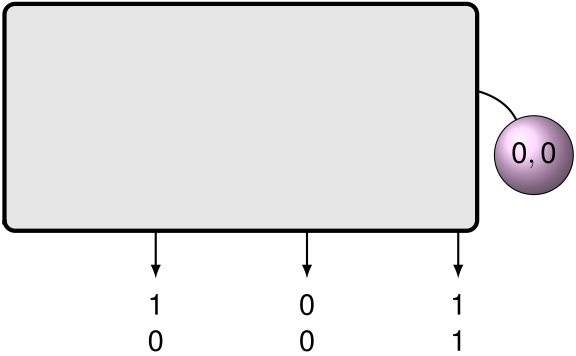
# Hidden Markov model



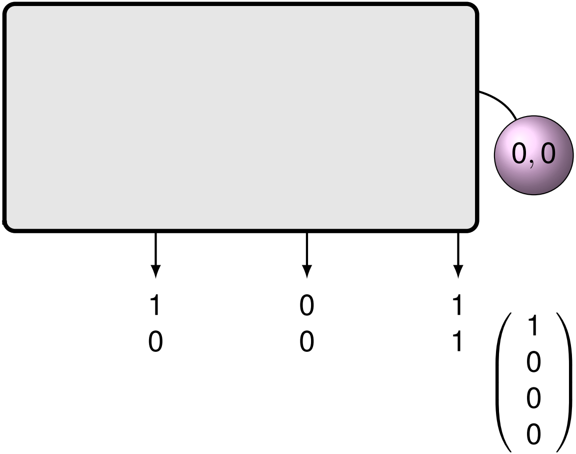
# Hidden Markov model



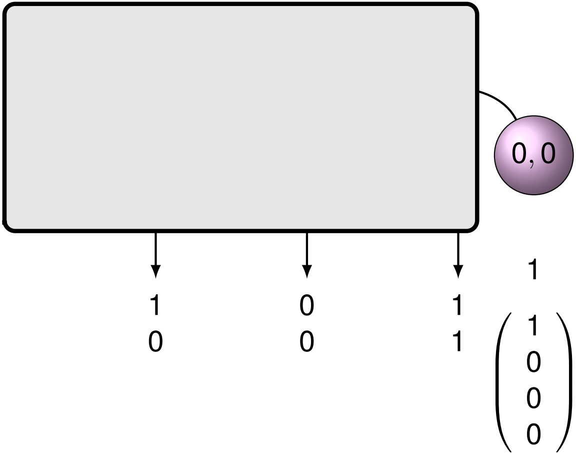
# Hidden Markov model



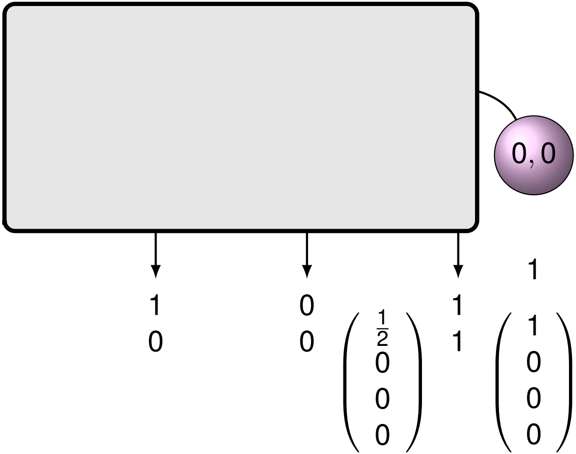
# Hidden Markov model



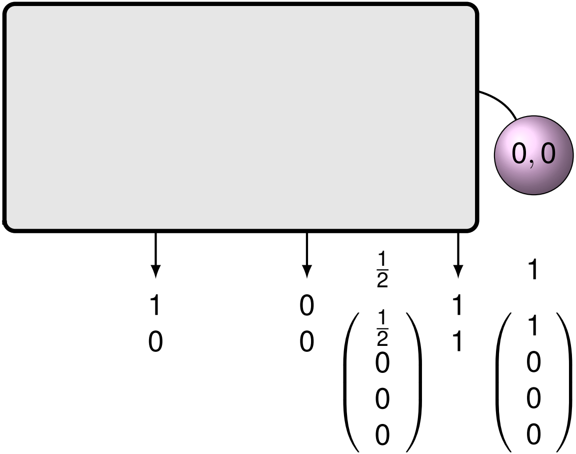
# Hidden Markov model



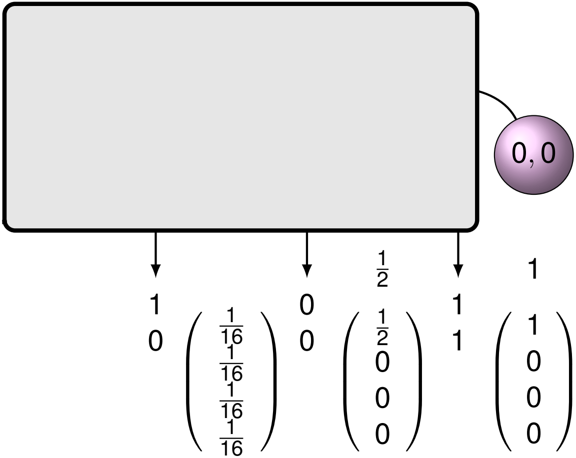
# Hidden Markov model



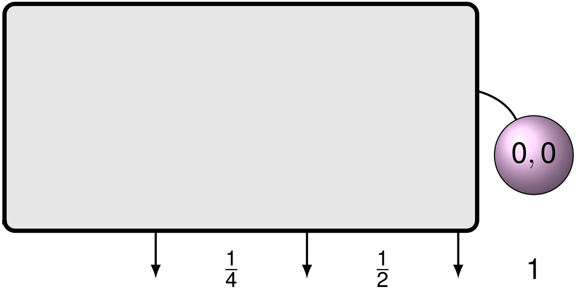
# Hidden Markov model



# Hidden Markov model

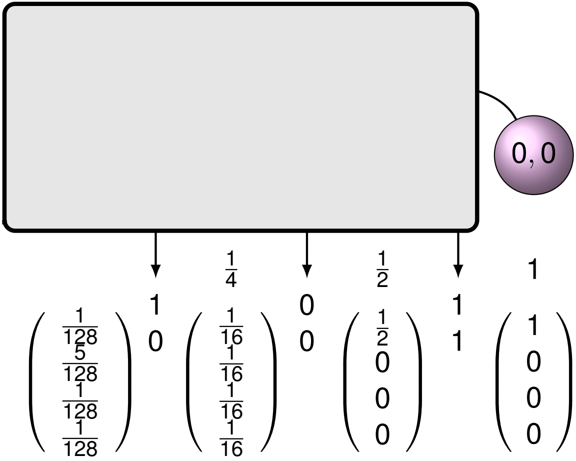


# Hidden Markov model

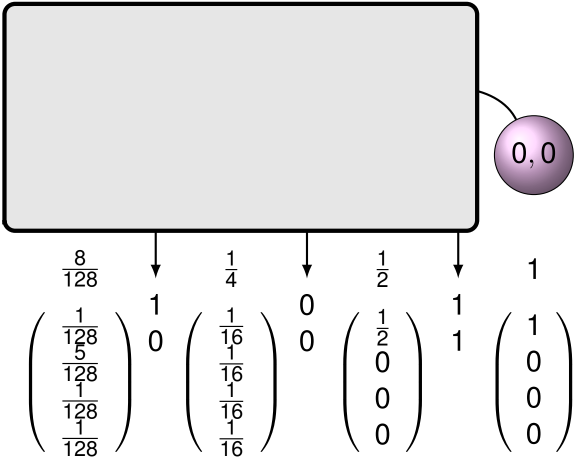



$$\begin{pmatrix} \frac{1}{16} \\ \frac{1}{16} \\ \frac{1}{16} \\ \frac{1}{16} \end{pmatrix} \quad \begin{pmatrix} \frac{1}{2} \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

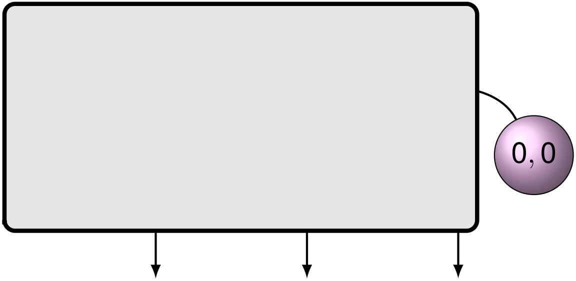
# Hidden Markov model



# Hidden Markov model



# Hidden Markov model




$$\begin{pmatrix} \frac{1}{128} \\ \frac{5}{128} \\ \frac{1}{128} \\ \frac{1}{128} \end{pmatrix} = \overset{\overset{1}{0}}{M_{1,0}} \quad \overset{\overset{0}{0}}{M_{0,0}} \quad \overset{\overset{1}{1}}{M_{1,1}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

# The forward algorithm

$$\Pr\left[\begin{array}{ccc} 1 & 0 & 1 \\ 0 & 0 & 1 \end{array}\right] = \mathbf{1}_{1\times 4} M_{1,0} M_{0,0} M_{1,1} \pi_0$$

where $\mathbf{1}_{1\times 4} = \left(\begin{array}{cccc} 1 & 1 & 1 & 1 \end{array}\right)$ and $\pi_0 = \left(\begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \end{array}\right)$

# Our attack

- For each sample, use the forward algorithm to find $\Pr[s|L]$
- from which we estimate $\Pr[s|\text{Py}]$
- We score each sample $s$ with $\mathrm{LLR}(s) = \log(\frac{\Pr[s|\text{Py}]}{\Pr[s|\text{Random}]})$
- Sum of scores is log-likelihood ratio for whole sample
- If score is positive, output Py
- Otherwise, output Random

# Efficacy of our distinguisher

$$\sum_{z \in \mathcal{Z}} \Pr[z|L]^2$$

# Efficacy of our distinguisher

$$
\begin{aligned}
& \sum_{z \in \mathcal{Z}} \Pr[z|L]^2 \\
= \; & \sum (\mathbf{1}_{1\times 4} M_{31} M_{30} \ldots M_0 \pi_0)^2
\end{aligned}
$$

$$M_i \in \{M_{0,0}, M_{0,1}, M_{1,0}, M_{1,1}\}$$

# Efficacy of our distinguisher

$$
\begin{aligned}
& \sum_{z \in \mathcal{Z}} \Pr[z|L]^2 \\
= & \sum \left(\mathbf{1}_{1\times 4} M_{31} M_{30} \ldots M_0 \pi_0\right)^2 \\
= & \sum \left(\mathbf{1}_{1\times 4} M_{31} M_{30} \ldots M_0 \pi_0\right)\left(\mathbf{1}_{1\times 4} M_{31} M_{30} \ldots M_0 \pi_0\right)^T
\end{aligned}
$$

$$
M_i \in \{M_{0,0}, M_{0,1}, M_{1,0}, M_{1,1}\}
$$

# Efficacy of our distinguisher

$$
\begin{aligned}
& \sum_{z \in \mathcal{Z}} \Pr[z|L]^2 \\
= & \sum \left(\mathbf{1}_{1\times 4} M_{31} M_{30} \ldots M_0 \pi_0\right)^2 \\
= & \sum \left(\mathbf{1}_{1\times 4} M_{31} M_{30} \ldots M_0 \pi_0\right) \left(\mathbf{1}_{1\times 4} M_{31} M_{30} \ldots M_0 \pi_0\right)^T \\
= & \sum \left(\mathbf{1}_{1\times 4} M_{31} M_{30} \ldots M_0 \pi_0 \pi_0^T M_0^T \ldots M_{30}^T M_{31}^T \mathbf{1}_{1\times 4}^T\right)
\end{aligned}
$$

$$M_i \in \{M_{0,0}, M_{0,1}, M_{1,0}, M_{1,1}\}$$

# Efficacy of our distinguisher

$$\begin{aligned}
& \sum_{z \in \mathcal{Z}} \Pr[z|L]^2 \\
= & \sum \left(\mathbf{1}_{1\times 4} M_{31} M_{30} \ldots M_0 \pi_0\right)^2 \\
= & \sum \left(\mathbf{1}_{1\times 4} M_{31} M_{30} \ldots M_0 \pi_0\right)\left(\mathbf{1}_{1\times 4} M_{31} M_{30} \ldots M_0 \pi_0\right)^T \\
= & \sum \left(\mathbf{1}_{1\times 4} M_{31} M_{30} \ldots M_0 \pi_0 \pi_0^T M_0^T \ldots M_{30}^T M_{31}^T \mathbf{1}_{1\times 4}^T\right) \\
= & \mathbf{1}_{1\times 4} \sum \left(M_{31} M_{30} \ldots M_0 \pi_0 \pi_0^T M_0^T \ldots M_{30}^T M_{31}^T\right) \mathbf{1}_{1\times 4}^T
\end{aligned}$$

$$M_i \in \{M_{0,0}, M_{0,1}, M_{1,0}, M_{1,1}\}$$

# Efficacy of our distinguisher

$$H_i = \sum M_{i-1} M_{i-2} \dots M_1 M_0 \pi_0 \pi_0^T M_0^T M_1^T \dots M_{i-2}^T M_{i-1}^T$$

# Efficacy of our distinguisher

$$
\begin{aligned}
H_i &= \sum M_{i-1} M_{i-2} \ldots M_1 M_0 \pi_0 \pi_0^T M_0^T M_1^T \ldots M_{i-2}^T M_{i-1}^T \\
H_0 &= \pi_0 \pi_0^T
\end{aligned}
$$

# Efficacy of our distinguisher

$$
\begin{aligned}
H_i &= \sum M_{i-1} M_{i-2} \ldots M_1 M_0 \pi_0 \pi_0^T M_0^T M_1^T \ldots M_{i-2}^T M_{i-1}^T \\
H_0 &= \pi_0 \pi_0^T \\
H_{i+1} &= \sum_{M \in \{M_{0,0}, M_{0,1}, M_{1,0}, M_{1,1}\}} M H_i M^T
\end{aligned}
$$

# Efficacy of our distinguisher

$$
\begin{aligned}
H_i &= \sum M_{i-1} M_{i-2} \ldots M_1 M_0 \pi_0 \pi_0^T M_0^T M_1^T \ldots M_{i-2}^T M_{i-1}^T \\
H_0 &= \pi_0 \pi_0^T \\
H_{i+1} &= \sum_{M \in \{M_{0,0}, M_{0,1}, M_{1,0}, M_{1,1}\}} M H_i M^T \\
\beta &= \Pr[L]^2 \left( 2^{64} \left( \mathbf{1}_{1\times 4} H_{32} \mathbf{1}_{1\times 4}^T \right) - 1 \right)
\end{aligned}
$$

# Efficacy of our distinguisher

$$
\begin{aligned}
H_i &= \sum M_{i-1} M_{i-2} \ldots M_1 M_0 \pi_0 \pi_0^T M_0^T M_1^T \ldots M_{i-2}^T M_{i-1}^T \\
H_0 &= \pi_0 \pi_0^T \\
H_{i+1} &= \sum_{M \in \{M_{0,0}, M_{0,1}, M_{1,0}, M_{1,1}\}} M H_i M^T \\
\beta &= \Pr[L]^2 \left( 2^{64} \left( \mathbf{1}_{1\times 4} H_{32} \mathbf{1}_{1\times 4}^T \right) - 1 \right) \\
&\approx 60552 \Pr[L]^2
\end{aligned}
$$

# Conclusions

- We can efficiently calculate the efficacy of HMM-based distinguishers
- Distinguisher advantage is 0.53 given $2^{64}$ bytes from $2^{8}$ key/IV pairs
- Advantage is 0.03 given a single $2^{64}$-byte stream
- Can this be improved still further?

*http://www.ciphergoth.org/crypto/py*