Mimicry: An Introduction
By Mystic
What is Mimicry?

- The ability to survive by mimicking surroundings
What is Mimicry?

- The ability to survive by mimicking surroundings
- Visibility of encryption
What is Mimicry?

- The ability to survive by mimicking surroundings
- Visibility of encryption
- Grammar filtering to find encrypted data
Peter Wayner’s Mimic Functions

- A way to encrypt/hide data in which the output is statistically and grammatically sound
Peter Wayner’s Mimic Functions

- A way to encrypt/hide data in which the output is statistically and grammatically sound.
- Generates text using the syntax described in a Context Free Grammar (CFG) and hides data by the choices it makes.
Context Free Grammars

- A way of describing language
Context Free Grammars

- A way of describing language
- Uses
  - Terminals: Words and phrases that are static
Context Free Grammars

- A way of describing language

- Uses
  - Terminals: Words and phrases that are static
  - Variables: Places where decisions can be made
Context Free Grammars

- Productions: describes how a variable can be converted into different sets of variables or terminals
Context Free Grammars

- Productions: describes how a variable can be converted into different sets of variables or terminals

- Example of a production:
  
  `variable --> phrase | | words`
Example of a CFG:

Start $\rightarrow$ noun space verb

noun $\rightarrow$ Fred $|$ $|$ Barney $|$ $|$ Fred and Barney

verb $\rightarrow$ went fishing. $|$ $|$ went bowling.
How Data is Hidden

- Mimic generates a binary tree based on the possibilities in a context free grammar
How Data is Hidden

- Mimic generates a binary tree based on the possibilities in a context free grammar

- It then chooses the leaves that would encode the right bits
How Data is Hidden

Given the following CFG:

Start ---> noun verb
noun ---> Fred || Barney
verb ---> went fishing where || went bowling where
where ---> in direction Iowa. || in direction Minnesota.
direction ---> northern || southern
How Data is Hidden

- Bits to be hidden: 1010

Start ---> noun verb

noun ---> Fred || Barney
verb ---> went fishing where || went bowling where
where ---> in direction Iowa. || in direction Minnesota.
direction ---> northern || southern

<table>
<thead>
<tr>
<th>Step</th>
<th>Answer in Progress</th>
<th>Bit Hidden</th>
<th>Production Choice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Start</td>
<td>none</td>
<td>Start ---&gt; noun verb</td>
<td></td>
</tr>
<tr>
<td>2 noun verb</td>
<td>1</td>
<td>noun ---&gt; Barney</td>
<td></td>
</tr>
<tr>
<td>3 Barney verb</td>
<td>0</td>
<td>verb ---&gt; went fishing where</td>
<td></td>
</tr>
<tr>
<td>4 Barney went fishing where</td>
<td>1</td>
<td>where ---&gt; in direction Minnesota.</td>
<td></td>
</tr>
<tr>
<td>5 Barney went fishing in direction</td>
<td>0</td>
<td>direction ---&gt; northern Minnesota</td>
<td></td>
</tr>
</tbody>
</table>

Final sentence: Barney went fishing in northern Minnesota
Mimic Grammar File

- Variables: Start with an asterisk and must be one word
Mimic Grammar File

- **Variables**: Start with an asterisk and must be one word
- **Productions**: are separated by numbers between forward slashes
Mimic Grammar File

- Variables: Start with an asterisk and must be one word
- Productions are separated by numbers between forward slashes
- The numbers indicate the weight given to the production (the higher the number the more probable it will occur in the production)
Mimic Grammar File

- **Variables:** Start with an asterisk and must be one word
- **Productions are separated by numbers between forward slashes**
- **The numbers indicate the weight given to the production (the higher the number the more probable it will occur in the production)**
- **The end of a variable is indicated by double slashes**
Mimic Grammar File

- The starting variable is the variable that is alphabetically first.
Mimic Grammar File

- The starting variable is the variable that is alphabetically first

- Example:
  
  *AAStart = Fred went to *con /.1/ Barney went to *con /.1//
  
  *con = Defcon /.1/ Black Hat /.01//
Mimic Grammar File

- Grammars can not be ambiguous, meaning there must be only one way of producing any given phrase (sentence)
Grammars can not be ambiguous, meaning there must be only one way of producing any given phrase (sentence)

Example of an ambiguous grammar:

```
*AAStart = *first *second

*first = love /.5/ love and /.5/

*second = death /.5/ and death /.5/
```
To avoid ambiguous grammars, grammars must be in Greibach Normal Form (GNF) meaning variables must be at the end of productions.
Limitations of Mimic Functions

- Can create huge files from small amounts of text, depends on the number of possibilities in the grammar
Limitations of Mimic Functions

- Can create huge files from small amounts of text, depends on the number of possibilities in the grammar
- Output starts repeating
Limitations of Mimic Functions

- Can create huge files from small amounts of text, depends on the number of possibilities in the grammar
- Output starts repeating
- Randomness is built in, but not very sophisticated
Mimicry in a live chat environment
ircMimic

- Mimicry in a live chat environment
- Sends a Mimic encoded message in the form of one side of an IRC conversation
ircMimic

- Mimicry in a live chat environment
- Sends a Mimic encoded message in the form of one side of an IRC conversation
- ircMimic Grammar File
  - Added “new line” character
  - Maximized possibilities
ircMimic

- Going further with ircMimic
  - Have two bots talk with each other in order to send a message
ircMimic

- Going further with ircMimic
  - Have two bots talk with each other in order to send a message
  - Add more to the grammar
Resources + Links

- **Mimic**
  - [http://www.wayner.org/texts/mimic/](http://www.wayner.org/texts/mimic/)
  - [http://www.spammimic.com](http://www.spammimic.com)

- **Disappearing Cryptography**
  - [http://www.wayner.org/books/discrypt2/](http://www.wayner.org/books/discrypt2/)

- **ircMimic**
Resources + Links

- TVSG
  - http://www.tvsg.org/