An Interactive Tool for Learning Linear and Differential Cryptanalysis of SPNs

Mike Carpenter
Rochester Institute of Technology, Computer Science Department

Linear and differential cryptanalysis (LDC) are techniques for analyzing Substitution-Permutation Network (SPN) ciphers by approximating certain qualities of their S-Boxes. Approximations are useful in discovering pieces of the cipher’s key influenced by approximated S-boxes.

INTRODUCTION

LDC doesn’t require knowledge of previous rounds’ keys, avoiding brute force. For a 5-round, 16-bit SPN, obtaining 8 bits of a subkey through linear cryptanalysis requires testing 256 (2^8) subkeys. Brute-forcing the cipher requires trying more than 1.2 septillion (2^{24}) keys.

An application to automate the process for a near-arbitrary SPN (within certain limits) does not currently exist, but would be useful, especially in classroom environments. The difficulty of computer science, and cryptography in particular, makes it a good target for interactive and visual learning applications.

Other cryptography visualization tools do exist, and the example set by their feedback of user choices. Other dialogs like the permutation definition dialog display instant feedback of user choices.

The goal of this project is to produce an application which can be used both to perform and to teach linear/differential cryptanalysis as a supplementary tool alongside classroom instruction.

The tool is a Swing utility written in Java which helps guide a user through the process of defining and cryptanalyzing a cipher.

Help documentation is included on relevant dialogs.

Users can serialize their SPNs to XML for later use or use as examples.

The application supports block sizes up to 64 bits and any S-box size which fits (though larger S-boxes may take too much screen space).

Can store a reference to a previously-defined component in a new part of the structure to prevent defining an identical component multiple times.

As opposed to the Heys tutorial, which uses a textual explanation with a static image, this tool allows a user to define their own approximation and watch the path of bits through the cipher appear instantaneously.

Other dialogs like the permutation definition dialog display instant feedback of user choices.

The underlying cipher structure:
- SBox – Integer array maps input values to output values.
- Permutation – Integer array maps input bit positions to output bit positions.
- Key – Holds a value and allows XOR operations with inputs.
- Round – Contains Key (subkey), set of SBoxes, and a Permutation; calls those objects in order.
- SPNetwork – A set of Rounds applied in order.

BACKGROUND

LDC doesn’t require knowledge of previous rounds’ keys, avoids brute force. For a 5-round, 16-bit SPN, obtaining 8 bits of a subkey through linear cryptanalysis requires testing 256 (2^8) subkeys. Brute-forcing the cipher requires trying more than 1.2 septillion (2^{24}) keys.

An application to automate the process for a near-arbitrary SPN (within certain limits) does not currently exist, but would be useful, especially in classroom environments. The difficulty of computer science, and cryptography in particular, makes it a good target for interactive and visual learning applications.

The tool created for this project is a successful first step toward a robust and versatile instructional cryptanalysis tool.

The tool can retrieve partial subkeys for both linear and differential cryptanalysis attacks.

Keys are not always totally accurate, although probability increases with more plaintext/ciphertext pairs and weaker S-boxes—this is consistent with what is expected from these cryptanalysis techniques.

The tool reports a variety of important information calculated about the cipher including biases, tables, and calculated subkeys.

FUTURE WORK

Ideally, the tool should be trialed in a classroom environment to obtain feedback from learners about its efficacy in conveying the essential concepts as compared to written tutorials.

A number of improvements could build upon the work done in this project:
- Create more detailed help material to further reduce or eliminate need for external reference material to understand concepts.
- Use results of classroom trials to incorporate more effective teaching strategies in the tool and its help material.
- “Unknown Key Mode” where pre-encrypted pairs are provided by user.
- Unify SPN visualization and definition tree onto two panes of single dialog.
- Refine UI elements to increase navigability, consistency, and clarity.

REFERENCES


ACKNOWLEDGEMENTS/CONTACT

Author: Mike Carpenter (mjc7806@g.rit.edu)
Repository: https://github.com/mjc7806/maledict
Advisor: Edith Hemaspaandra (eh@cs.rit.edu)