

# The Design and Use of Interactive Visualization Applets for Teaching Ciphers

Dino Schweitzer, Leemon Baird

**Abstract**—Cryptography is a fundamental topic in an information assurance curriculum. Students should understand the basic concepts and weaknesses of both historical and current cipher algorithms. Visualization tools can help students understand these concepts, both in the classroom and as out-of-class exercises. This paper describes a set of such tools designed for a Cryptography Course at the United States Air Force Academy. The design goals, implementation details, and classroom experiences are addressed.

**Index Terms**—Information Assurance Education, Cryptography, Classroom Visualization

## I. INTRODUCTION

A key priority in the National Strategy to Secure Cyberspace is to increase security training and awareness through enhanced education programs [1]. To encourage security education, increase research, and produce a growing number of information assurance professionals, the National Security Agency in conjunction with the Department for Homeland Security created and administers the Centers of Academic Excellence (CAE) in Information Assurance Education Program [2]. Schools with the CAE designation must meet strict guidelines in educational and research excellence.

As a result of this emphasis on security education, a greater number of courses and programs are being offered at the undergraduate level in security-related topics such as cryptography, information security, network security, and information warfare. These courses have benefited from an increasing number of textbooks, curriculum development, and student competitions such as the Cyber Defense Exercise [3]. These educational resources provide a solid foundation for developing a series of courses in information assurance education. Such courses can be further enhanced by the availability of additional education tools. Educational concept visualization, or pedagogical visualization, is one approach that can greatly benefit security education.

Cryptography is a core topic in information assurance and

has been taught in many programs for several years. One of the fundamental concepts within cryptography is the subject of ciphers, how information is encrypted and decrypted and how easy or difficult it is to break a specific cipher. Ciphers also have a rich historical background that makes them attractive to a broad range of students. Some students find the basic concepts of how different ciphers operate difficult to grasp based on textual description and mathematical notation alone. For these students, a hands-on visual approach can be more effective in communicating the basic concepts of the cipher's details.

## II. PEDAGOGICAL VISUALIZATION

Visualizations of abstract concepts for purposes of education have been effective in several disciplines for many years. In computer science, algorithm and data structure visualizations have been around since the 1980's [4]. In mathematics, advances in computer graphics led to visualizing mathematical processes as well as objects [5]. Software tools such as Maple, Mathematica, and Matlab have made mathematical visualization available as a learning aid to all levels of education [6]. In the physical sciences, molecular visualization and scientific visualization have aided student understanding of both abstract and physical processes [7]. Numerous web sites provide interactive visual demonstrations to assist in teaching complex concepts in many disciplines.

One key distinction between different types of educational visualizations is the underlying level of abstraction. Visualization of a physical object such as a water pump to demonstrate its operation can be accomplished with a very low level of abstraction such as an image of the pump. Visualization of a data structure or algorithm, however, requires a graphical representation of non-physical objects and the operations performed on them.

Discipline-specific research has been conducted as to the effectiveness of visualization in the educational process. For example, a number of studies have evaluated the effectiveness of algorithm animations. While the studies are not universal in agreement as to the overall success of these approaches, a key finding of a meta-study of these evaluations is that the most successful educational uses of visualization technology are those in which the technology is used as a vehicle for actively engaging students in the learning process [8]. In other words, static images or simple pre-computed animations are insufficient to leverage the power of a visualization

Manuscript received March 22, 2006.

Dino Schweitzer is Director of the Academy Center for Information Security at the United States Air Force Academy, CO 80840 USA (phone: 719-333-3945; fax: 719-333-3338; e-mail: dino.schweitzer@usafa.af.mil).

Leemon Baird is with the Computer Science Department at the United States Air Force Academy, CO 80840 USA (e-mail: leemon.baird@usafa.af.mil).

approach. Rather, the visualization should allow the student to actively interact with it, modifying parameters and stepwise controlling execution to perform “what-if” analyses and predictive exercises.

Some researchers in educational psychology believe the process of how students learn with computer-based learning aids is not well understood [9]. As a result, tool development is often based more on what computers can do rather than applying well-defined pedagogical principles. A working group formed to review the effectiveness of visualization in computer science education identified a set of “best practices” in pedagogical visualization based on published studies and personal experiences [10].

Additional studies have been performed on the most effective means of utilizing tools once they have been created. Options include, but are not limited to: classroom demonstration, open and closed laboratory exercises, as part of online e-Learning modules, as part of hypertext documents, and as open resources on web sites. Today, some form of web-based tool, usually a Java applet, is the most prevalent approach to facilitate cross-platform access and provide a great deal of flexibility in how the visualization is utilized in a course.

Some developers have focused on creating systems and building blocks allowing educators and students to create their own visualizations. Polka, Samba, and XTango are examples of general-purpose systems developed at Georgia Tech for creating algorithm animations and software visualizations [11]. For visualization of concepts in computer graphics, Brown University created a basic set of Java applets that could be used for developing visualizations of more sophisticated concepts [12].

### III. CRYPTOGRAPHY VISUALIZATIONS

Visualization and interactive demonstrations are not a new concept in the field of cryptography. Several web sites host a variety of different approaches to different cryptography concepts and ciphers. A common choice for visualization has been the Enigma machine with several websites hosting visual simulations of the famous machine [13]. Some approaches focus on an animation approach that demonstrate the steps of the algorithm in a visual manner while others allow the user to type in text and key information and perform the encryption/decryption. An example of both approaches can be found at Embry-Riddle’s web site with the work of Susan Gerhart [14]. Concepts such as DES key generation and diffusion are demonstrated with animated applets showing the sequence of bit-manipulation steps. The visualization incorporates minimal user interaction other than stepping through the animation. Other concepts, such as an MD5 demonstration, involve more user interaction allowing them to enter arbitrary text and step through the encryption process. Other examples are the cryptography applets used at Central College [15]. The user can choose different algorithms to see arbitrary text encrypted/decrypted. Additional cryptographic

tools such as letter frequency and an anagram analyzer are available. The approach of using Java applets to demonstrate cryptography has been used by one author to complement a textbook on the subject [16]. Of the cryptography tools reviewed, perhaps the most comprehensive interactive one is the freeware program CrypTool developed in Germany [17]. This tool is a robust standalone program with a wide variety of both classical and modern encryption techniques including several tools for analyzing the various algorithms.

### IV. CIPHER VISUALIZATIONS AT USAFA

At the Air Force Academy, we undertook a project to create a set of cipher visualizations to support the undergraduate course in Cryptography taken primarily by Computer Science and Mathematics majors. The primary purpose of the project was to produce interactive visual demonstrations of various encryption algorithms and code-breaking concepts to support in-class lectures. While some of the existing visualizations on the web contained different aspects of concepts we wanted to demonstrate, there was not a single comprehensive set that met our needs. We wanted the tools to be highly interactive beyond simple animations or merely inputting plaintext and key information. For example, sliding a slider to interactively see how frequencies diagrams are changing. We also wanted the visualizations to have a strong visual component with extensive use of diagrams, colors, and graphs. One of the most important requirements was that the visualizations had to be quickly understood in a classroom setting to be effective. Tools such as CrypTool require a learning curve to fully understand what is happening and to effectively use it. Finally, we wanted the tools to add an element of “fun” to the classroom. The following sections describe the cipher selection, design goals, and implementation of the visualizations.

#### A. Cipher Selection

The ciphers for the visualization project were chosen from both historical and current ciphers to emphasize key points in the operation and evaluation of cipher algorithms. For example, the Vigenère cipher was chosen due to its historical prominence and widespread belief in its invulnerability for hundreds of years. The fact that students can break the cipher with relative ease underscores a sense of skepticism when looking at current vendor claims. The following ciphers were chosen to be represented:

- Shift Cipher
- Simple Substitution Cipher
- Affine Cipher
- Vigenère Cipher
- RC4 Stream Cipher
- RSA Cipher
- DES Cipher

### B. Design Goals

The following design goals for the set of cipher visualizations are based on the experiences described in the literature in algorithm visualizations.

#### 1) High Interactivity

Each visualization has user-controlled input for the input text to be encrypted as well as the key. This allows the user to make small changes to the environment and see the results of the change. For example, does changing a single letter in the input text affect a single character or multiple characters of the output ciphertext? Or, in the case of the affine cipher, how a simple change in key selection can result in multiple letters encrypting to the same value.

In addition to the selection of text and key, users have interactive control over cipher analysis tools such as frequency graphs, key length analysis, and digram maps. Extensive use of GUI devices, such as buttons, sliders, and spinners are used to interact with the visualizations.

#### 2) Consistency of Representation

An important consideration when using visualization tools in the classroom is the amount of time required for the student to assimilate the abstract representation being presented. If the representation is too obscure, or changes from visualization to visualization, precious time is wasted from understanding the concepts being presented. This design goal is especially important when presenting a family of visualizations on the same subject matter such as ciphers.

To meet this design goal, the same basic layout was used in all of the ciphers tools developed as shown in Figure 1. The upper panel demonstrates the cipher encryption while the lower tabbed panels demonstrate specific aspects of that particular cipher. In the upper panel, the top line is the user-supplied input text to be encrypted followed by the key information which changes slightly from cipher to cipher. Following the key is a set of control buttons for the basic cipher operations. Once again, these vary slightly from cipher to cipher. Next is the plaintext and encrypted ciphertext along with any descriptive elements of how the ciphertext was generated. For example, in Figure 1, the Key Offset and Shift amounts are shown as part of the Vigenere ciphertext generation. Figure 2 shows the fields after a plaintext and key have been entered. Figure 1 shows an example of the tabs for the Vigenere cipher applet (Key Length and Frequency) which open lower panels for showing more detailed information and cipher breaking strategies unique to that cipher. Other examples of cipher-specific tabs are digrams and single letter frequencies for the substitution cipher and an affine map and letter frequency for the affine cipher.

Most visualizations have both the *Encrypt* and *Decrypt* buttons to show the process of how the ciphertext is created and interpreted. In addition, several of the ciphers have the *Secret* button which randomly selects text and shows the ciphertext, but neither the plaintext nor the key. This allows

the instructor to demonstrate how to break the cipher and generate the plaintext using a known attack.

Another element of consistency was using the same visual representation for common elements such as the frequency diagram always showing both English and ciphertext frequencies in the same format. This can be seen in the upper bar chart of Figure 4.



Fig. 1. Basic layout of cipher applets.



Fig. 2 Upper panel with plaintext and key entered.

#### 3) Minimize Keyboard Interaction

This design goal was based on our experience in using these visualizations in the classroom. When the instructor is projecting a visualization to explain a concept and the lights are dimmed for projection purposes, keyboard interaction is awkward and error-prone. While some keyboard input is still required to input the initial text and key information, further interactions are primarily by mouse interaction. For example, when identifying a letter in the Vigenère cipher using frequency analysis, the letter can be added to the key through the *Add to Key* button and does not require a separate key press.

#### 4) Standalone Descriptive Text

This goal refers to the desire for the visualizations to be available for use in either an instructor-led scenario such as a classroom or lab, or by the student alone outside of the classroom. Sufficient information needs to be presented in order for the student to understand how the cipher works, what the various controls in the visualization do, and what they are showing about the cipher's characteristics. For example, what does it mean when the affine map shows two letters encrypting to the same value?

To accomplish this goal, we developed each visualization as a Java applet embedded on a web page with explanatory text. The student/instructor can use the applet and ignore the text, or read the text along with the applet's use. This also allows the instructor to embed the applet in a different web page for different applications, such as an out-of-class exercise with specific text instructions.

5) Algorithm representation and execution

The concepts of abstraction, consistency, and interactivity are especially important when demonstrating algorithms. When an applet describes the execution of an algorithm, such as the key generation phase of RC4, the algorithm is represented by a small number of high-level pseudocode steps as shown in Figure 3. Execution of the algorithm is accomplished through a set of control buttons allowing the algorithm to step forward or backward, resetting, or completing. An arrow is used to indicate which line of pseudocode is the current line executing. In the example shown, each step will update the S-Array with the appropriate visual indicators as to what the current location is, what values were swapped, and what the current key character is. Single step control and the ability to back up are important for students to recognize what event just occurred and be able to back up and replay the step.

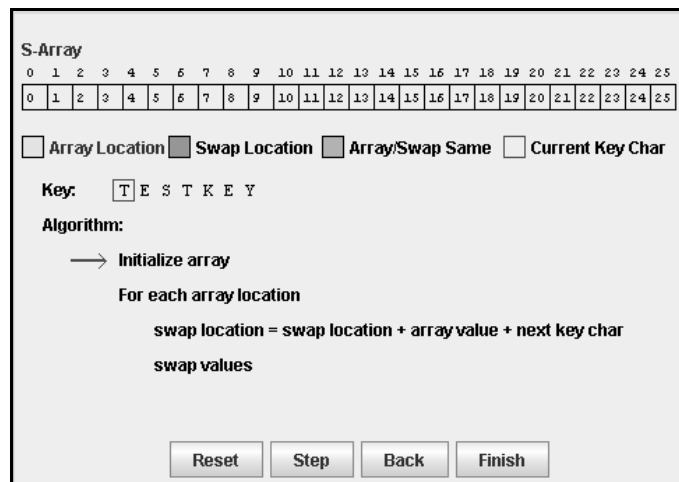


Fig. 3. Key matrix initialization algorithm for RC4.

C. Implementation

As previously stated, each cipher visualization tool is implemented in Java as an applet. Standard Swing components are used for interactivity. The applets are embedded in explanatory web pages and made available to students. Each applet is under 50K in size avoiding unusually long delays when downloading across the net.

D. Cipher Tools Key Concepts

For each of the ciphers visualized, key concepts about the cipher are highlighted in the visualizations. A brief summary

of the specific concepts are described.

1) Shift Cipher

This is the simplest cipher illustrated, and is primarily used to familiarize the students with the basic layout and functions of the visualization tools. The basic layout is similar to Figure 1 with a place to enter input text and a shift key. When the user selects the *Encrypt* button the text is encrypted with the shift amount and displayed. The only cipher-specific tabbed panel is a frequency histogram as shown in Figure 4. This quickly demonstrates how easily the cipher can be broken. By sliding the shift amount using the slider bar until the frequencies best match up, the correct shift amount can be found. The second graph shown, the Frequency Dot Product displays the dot product of the vector formed by English language frequencies with the vector formed from the ciphertext frequencies at different shift amounts. This demonstrates to the student how to easily pick out the “best fit” for the correct shift amount visually by the “spike” in the graph. In the example shown, sliding the slider to a shift amount of nine will provide the best fit. If the ciphertext was generated by selecting the *Secret* button on the upper panel, the student can enter nine as the key shift and select the *Decrypt* button to see the hidden plaintext appear.

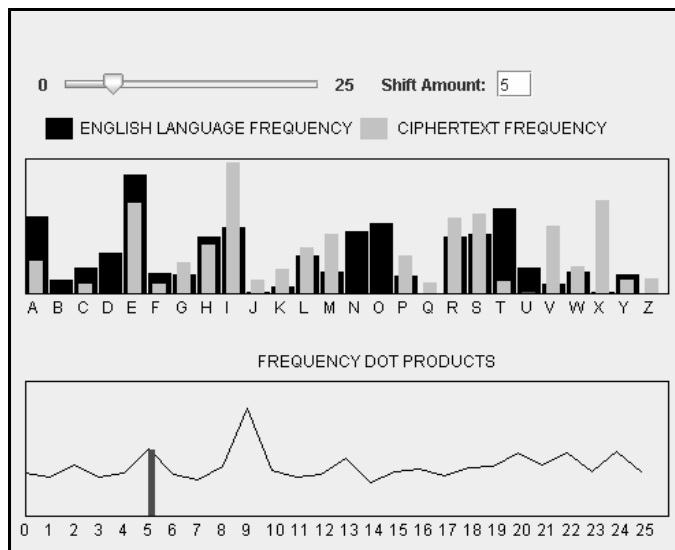


Fig. 4. Frequency tab of the shift cipher applet.

2) Simple Substitution Cipher

The simple substitution cipher is familiar to students as the typical newspaper “Cryptoquote”. Besides the standard encryption and decryption techniques, the tool highlights the use of single-letter and two-letter frequencies (digrams) to solve them. The digrams tab is shown in Figure 5 for the generated ciphertext. The instructor illustrates the use of these solving tools by selecting a “secret” plaintext and using the various frequencies to discover common letters such as *e*, or letter pairs, such as *th*. The applet allows partial substitution keys to be entered. When the user selects *Decrypt* the

partially solved plaintext is shown making full solution easy to complete.

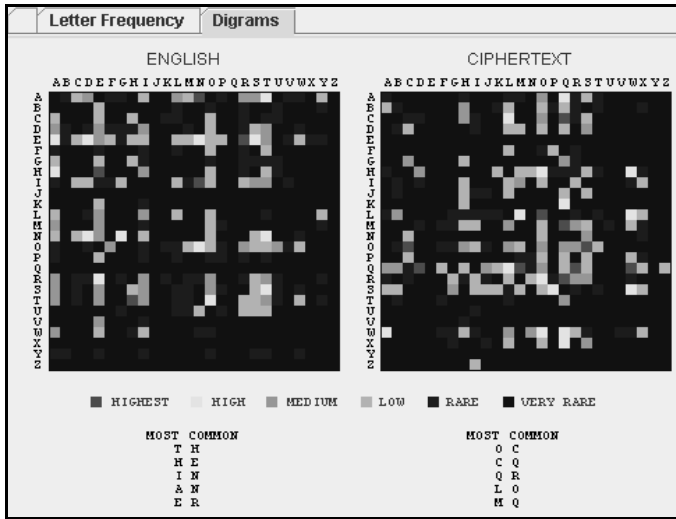


Fig. 5. Digrams tab in substitution cipher applet.

### 3) Affine Cipher

In the affine cipher applet in addition to the frequency tab, an Affine Map tab is included to show how letters are mapped between the plaintext and ciphertext for the chosen affine values of alpha and beta. This demonstrates to the student how the wrong choice (non-invertible) results in multiple letters mapping to the same ciphertext letter. Breaking an affine cipher is demonstrated in the frequency tab by sliding through all possible invertible alpha/beta pairs to find the best frequency match between the ciphertext and plaintext, as shown in Figure 6. When the selected Alpha and Beta are set in the upper panel, the “secret” message can be decrypted.

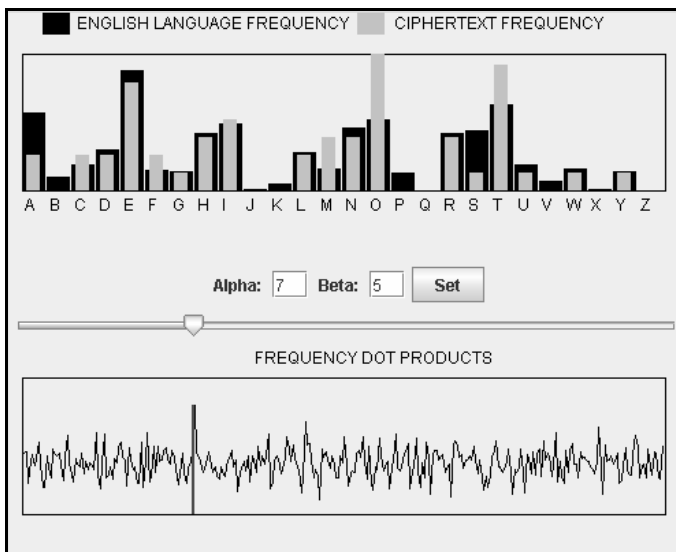


Fig. 6. Breaking an affine cipher by exhaustive search.

### 4) Vigenère Cipher

The upper panel for the Vigenère cipher applet is shown in Figure 1. This cipher is emphasized in the classroom as an example of a cipher that was used and thought secure by many for hundreds of years, but can now be shown to be quickly broken. The Key Length tab as shown in Figure 7 is used to show one technique for determining the key length. The cipher text is simply shifted on top of itself, and matched letters are counted to find the shift with the most matches. Once the key length is known, the cipher can be divided into several independent shift ciphers to solve. That is, for each letter in the key, the student can perform a frequency analysis using the sliding histogram and dot product graph as shown in Figure 4. The frequency tab of the Vigenère applet incorporates the same user interface that was used for the shift cipher, so students can use it without a long learning curve.

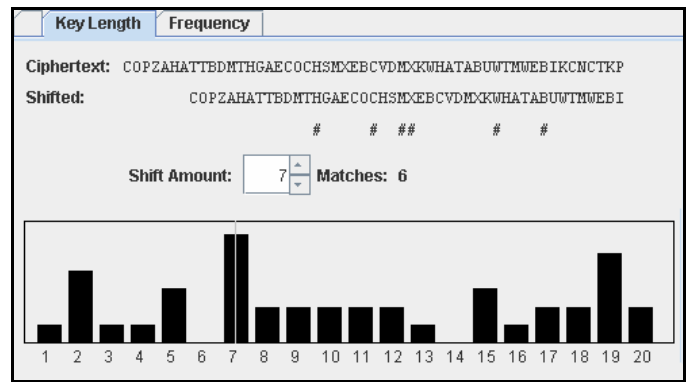


Fig. 7. Finding Vigenère key length by shifting ciphertext.

### 5) RC4 Stream Cipher

RC4 is used as an example of a stream cipher: it generates a stream of random bytes that are used to encrypt the data. Although it is widely used in certain applications (e.g. for secure websites), it has known problems. The applet interactively demonstrates the algorithms for key matrix initialization, as shown in Figure 3, as well as the algorithm for generating the characters for the key stream. As a simplification of the standard RC4 algorithm, the key matrix is composed of only 26 locations representing the letters of the alphabet.

### 6) RSA Cipher

RSA is the most famous and widely-used public-key cipher. This applet demonstrates the concept of breaking the input text into binary blocks for encryption. For demonstration purposes, two-character (16-bit) blocks are used. The user selects values for  $p$  and  $q$  to generate both the public key ( $d, n$ ) and the private key ( $e, n$ ).

### 7) DES Cipher

DES (and its triply-encrypted variant) is the oldest and most respected cipher in widespread use. Similar to the RSA cipher applet, the input text is broken into binary blocks for encryption. In addition, one of the applet tabs presents an

interactive visual demonstration of the DES encryption rounds. Two rounds of a Feistel system are illustrated as shown in Figure 8. The text accompanying the applet explains the purpose of the swapping, describes the  $f_n$  functions, and discusses how the key is used. The applet interactively moves bits through the diagram to show students the general flow of data through this type of cipher system.

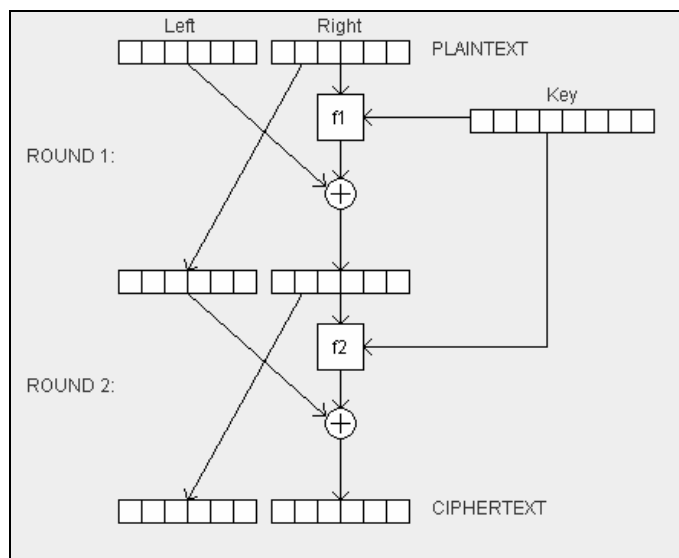


Fig. 8. Flow of bits through two rounds of DES.

## V. CLASSROOM EXPERIENCE

### A. General Use

These teaching tools can be used in four different approaches: 1) classroom instructor demos 2) classroom student exercises 3) web-based student learning resources 4) web-based student graded assignments.

As a classroom demonstration, the applets can be projected onto a screen where the students can watch. The instructor can first go through the steps of encryption or cryptanalysis on the screen, showing the students how to encrypt or break a cipher. The instructor can then go back and show "what if" scenarios, changing the parameters and showing how they affect the system.

As a student exercise, the applets can be used by the students on laptops or workstations in the classroom. The students can experiment with changing the settings and seeing how they affect the system. This active experimentation can build intuition for how the system works, often in ways that are better than passively watching a prerecorded video or animation.

As a web-based learning resource, the applets can be embedded in web pages that contain explanatory text. This can be a supplement to the course textbook, and can help students who need additional help on a given issue. Students can be required to read the pages as part of the assigned

reading, or can be given the pages as optional reading for additional help.

As a web-based graded assignment, the students can be required to experiment with the applet, then turn in written descriptions of what they have learned, or of answers to general questions about how these systems work, and how various changes might affect them.

### B. Sample Scenario

The power of these teaching tools is best illustrated with an example of how they were used in the undergraduate cryptography course at USAFA. In the course, the students were taught how the Vigenère cipher worked and how to break it. They then wrote programs to automatically break the cipher without human involvement.

The subject was first taught with traditional techniques, telling the students the list of steps to carry out, projecting useful illustrations on a screen, writing examples on the board, and having them work out small problems on paper at their seats. It was found that two subjects in particular confused the students: using the coincidence index to find the key length, and using a probability dot product to break each shift cipher Subproblem.

An additional lesson was then given that used the Vigenère applet. Every student owns a laptop computer, and they were asked to bring them to class. The applet was displayed on a large screen at the front of the room, while the students simultaneously ran it on their own computers. Figures 1 and 7 show the upper panel and Key Length panel for the applet used for this lesson.

The instructor illustrated how the spinner control in Figure 7 could be used to slide the ciphertext against itself. As the students experimented with it, they could see how a strong pattern emerged when they slid by a multiple of the key length. This gave them an intuitive, visual feel for how the key length could be found.

The instructor then took them to the Frequency tab (shown in Figure 4), which allowed the students to solve for the shift in every fifth letter. The histogram shows how frequent each letter is, both in the encrypted message and the message after decrypting with a given shift. By moving the slider, the student can see the effects of different shifts. The instructor was able to experiment with the shift in front of the students, and talk about how some shifts are better than others. In some cases, a shift might make the tallest bar line up well (which means it would make the frequency of "e" the most common), but would do a very bad job of making all the other bars line up. On the other hand, a different shift might make "e" slightly worse, but all the others much better. The students immediately grasped why it was not sufficient to just look at the tallest bar. The dot product is graphed in the second window, and as they tried different shifts, they could see how their position on that graph moved. This gave them an intuitive feel for how the dot product is highest when *all* the bars are matching well, not just the tallest bar for the letter "e".

### C. Student Reaction

The student reaction to this demonstration was very positive. First, they appeared to be more engaged, because they were performing experiments on the system themselves, rather than simply watching the instructor write equations on the board. Second, when questioned, they showed far more insight into why every fifth character has the pattern that it does, and why the dot product calculation is better than simply matching the most frequent character. This was very encouraging, since they had been confused on this point prior to using the applet.

It appeared that both aspects of the applet were important in giving the students true insight into the problem. First, it made abstract concepts concrete and visible. Second, it allowed the students to directly interact with the system, and get immediate feedback on the effects of changes to key length and possible shifts. This direct experimentation with immediate feedback seemed to allow some students to grasp subtle concepts that had eluded them when they were merely shown equations and lectured on the ideas.

## VI. CONCLUSION

Visualization tools can be an effective and entertaining means to teach students abstract concepts in the classroom. Providing accessibility to the tools outside of class allows motivated students to “play” with the visualizations. The tools can also be used for out-of-class assignments.

The cipher visualization tools described in this paper are part of a suite of visualization tools for security education being developed at the Air Force Academy known as VISE (Visualization for Information Security Education). The purpose of VISE is to provide a set of education visualizations for teaching undergraduate information security classes. The tools cover common security topics such as cryptography, buffer overflow, network attacks, and public key infrastructure. Source code for the cipher applets is publicly available at <http://usafa.af.mil/acis>. We welcome comments and feedback on user experiences.

## REFERENCES

- [1] Critical Infrastructure Protection Board, "National strategy to secure cyberspace", The White House, Washington, DC, USA. September 2002. Available: <http://www.whitehouse.gov/pcipb>
- [2] The National Centers of Academic Excellence in Information Assurance Education Program. 2005. Available: <http://www.nsa.gov/ia/academia/caeiae.cfm>
- [3] Haynes, A. and Stratton, T. "Cyber defense 2003 & information assurance education," *2003 IEEE International Conference on Systems, Man & Cybernetics*, Oct 2003.
- [4] Brown, M. and Sedgewick R. "A system for algorithm animation," *Computer Graphics*, pp.177-186, July 1984.
- [5] Palais, R. "The visualization of mathematics: towards a mathematical exploratorium," *Notices of the AMS*, vol. 46 no. 6, pp. 647-658 June/July 1999.
- [6] Chonacky N. and Winch D. "3Ms for instruction: reviews of Maple, Mathematica, and Matlab," *Computing in Science & Engineering*, Vol. 7 no.3, pp. 7-13, 2005.

- [7] Report from the Molecular Visualization in Science Education Workshop, NCSA Access Center, Arlington, VA, January 2001.
- [8] Hundhausen, C., Douglas, S., and Stasko, J. "A meta-study of algorithm visualization effectiveness," *Journal of Visual Languages and Computing*, vol. 13, pp. 259-290, 2002.
- [9] Mayer, R. and Sims, V. "For whom is a picture worth a thousand words? Extensions of dual-coding theory of multimedia learning," *Journal of Educational Psychology*, vol. 86, pp. 389-401, 1994.
- [10] Naps, T., Rößling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., Korhonen, A., Malmi, L., McNally, M., Rodger, S., and Velázquez-Iturbide, J. "Exploring the role of visualization and engagement in computer science education," *Working Group Reports From ITiCSE on innovation and Technology in Computer Science Education*, pp. 131-152, June 2002.
- [11] Graphics Visualization and Usability Center, Georgia Institute of Technology, Available: <http://www.gvu.gatech.edu/>
- [12] Brown University, Exploratories, Available: <http://www.cs.brown.edu/exploratories/freeSoftware/home.html>
- [13] Schwager, Russell, The Enigma Machine applet available at <http://russells.freeshell.org/enigma>
- [14] Gerhart, S. L. Increasing Security Expertise in Aviation-Oriented Computing Education: A Modular Approach. <http://nsfsecurity.pr.erau.edu>. 2005.
- [15] Linton, T., Cryptography web page available at: <http://www.central.edu/homepages/LintonT/classes/spring01/cryptograpy/cryptography.htm>
- [16] Bishop, D. Introduction to Cryptography with Java Applets, Jones and Bartlett Publishers, Boston, 2003.
- [17] Deutsche Bank AG, CrypTool , available at: <http://www.cryptool.org>