SHAvisual: A Visualization Tool for the Secure Hash Algorithm

Abstract

Data security is vital to the world we live in, and the foundation that secure communication is built upon is cryptography. Cryptography is a course that is regularly offered at colleges and universities. In our experience, computer science students find that understanding the sophisticated mathematics behind the crypto-systems is a daunting task, while math majors often get lost in the details of the complicated algorithms. Educators need to find a way to help students understand both what the algorithm does and why the algorithm does it. Visualization tools can be an effective way for educators to battle this challenge. This paper describes a tool, **SHAvisual**, which addresses this issue for the secure hash algorithm (SHA). SHA is a family of cryptographic hash functions that the National Institute of Standards and Technology began publishing in the early 1960's. **SHAvisual** is designed to help students learn and instructors teach the SHA-512 algorithm. It consists of three major components: Demo Mode, Practice Mode and Full Mode. A separate global view window helps highlight the current procedure in the algorithm pipeline. The Demo Mode provides a simplified SHA-512 visualization and is useful for the instructor to demonstrate important operations in the classroom. The Practice Mode is designed for students to learn the detailed computations step by step and perform self-study. A test report system also helps the instructor verify the learning effectiveness. The Full Mode is a full version of the SHA-512 cipher.

SHAvisual was classroom tested in a Computer Security course. Major findings of this survey include the following: (1) students indicated **SHAvisual** helped them understand SHA better; (2) both the Demo Mode and Practice Mode had positive impact on student learning; and (3) a MANOVA analysis suggested that the student reactions were generally independent of the time spent on using the software. Write-in comments also verified that **SHAvisual** did help students learn and the instructor teach the SHA algorithm effectively.

1. Introduction

Data security is vital to the world we live in, and the foundation that secure communication is built upon is cryptography. Modern cryptography research started in the late 1960's, and it has now grown into a significant field in math and computer science. Cryptography is a course that is regularly offered at colleges and universities. Textbooks and handbooks[8,10,13] aid in the teaching of cryptography, but the demographics of the students attracted to this field may present some unique challenges to educators. In particular, CS students find that understanding the sophisticated mathematics behind the crypto-systems is a daunting task, while math majors often get lost in the details of the complicated algorithms. Educators need to find a way to help students understand both what the algorithm does and why it does it.

Visualization tools can be an effective way for educators to battle this challenge. While some visualization tools have been developed, not many of them allow the user to understand both the mathematical theory and the algorithm behind a certain crypto-system. This paper describes a tool, **SHAvisual**, that does just that for the secure hash algorithm (SHA). SHA is a family of cryptographic hash functions that the National Institute of Standards and Technology began

publishing in the early 1990's. **SHAvisual** is a flexible tool that can be used both for demonstrative purposes in class and for students to explore the SHA-512 algorithm on their own. It is part of a larger project[9,15,16,17] which provides a number of visualization tools for teaching and learning cryptography. **SHAvisual** was evaluated in a computer security course and the results showed that **SHAvisual** helped students learn and the instructor teach the SHA algorithm effectively.

In the following, Section 2 discusses recent work and compares **SHAvisual** with other tools. Section 3 gives the background of the course in which we used **SHAvisual**, Section 4 presents our tool, Section 5 is a detailed study of our findings from a classroom evaluation and a self-study survey, and Section 6 has our conclusions.

2. Related Work

Software tools for demonstrating cryptography algorithms have started to appear in recent years. Some of these tools are rather complex for beginners[2] or have only limited user interactions[1,3,4]. Although SHA plays an important role in cryptography, not many visualization tools include SHA. While GRACE[7] offers a visualization framework for several cryptographic and non-cryptographic related operations (e.g., DES and RSA), it does not contain SHA. Schweitzer and Baird[13] developed several symmetric and asymmetric cipher visualization tools for a cryptography course without SHA. The SHA-1VA[11] is a web-based applet and the authors claimed that it is the first interactive software to demonstrate the SHA-1 algorithm step by step. **SHAvisual** and SHA-1VA are significantly different. SHA-1VA was developed for SHA-1 while **SHAvisual** is for a more advanced SHA-512 algorithm. Furthermore, the two tools differ in style and function. **SHAvisual** shows each component of the algorithm individually with detailed graphical presentations while SHA-1VA uses pseudo-code in a step-by-step format. **SHAvisual** has three separate modes: Demo, Practice, and Full. The Demo and Practice modes present the algorithm in a simplified way by reducing the size of blocks over which the computation is performed and removing duplicate operations. The Full mode performs the full SHA-512 algorithm, but shows only selected intermediate results. On the other hand, SHA-1VA combines the long hash value calculation with the demonstration all in one phase and does not include a practice mode. Thus, while SHA-1VA takes a more streamlined approach, **SHAvisual** can be used for assessment purposes as well as for learning. In particular, the Practice mode in **SHAvisual** helps users study the algorithm independently, and also provides instructors a way to measure teaching effectiveness. Details of the SHA-512 will be discussed in Section 4.

3. Course Information

SHAvisual was used in a computer security course, CS4471 Computer Security, that was offered out of the Department of Computer Sciences at Michigan Technological University. It is a senior level course that gives a basic introduction to topics in computer security. The cryptography component includes primitive ciphers, DES, Diffie-Hellman key agreement, RSA, and SHA. The course also covers secure coding in C, key management, authentication, access control, malicious logic, and intrusion detection.

Students in the course typically are computer science majors taking the course as an elective. The evaluation class included three Computer Systems Science majors who are required to take the course. All others in the class took it as an elective including 17 CS majors, three software engineering majors, seven electrical and computer and electrical engineering majors, and three students from other majors.

The SHA component was added after receiving questions from students on the operation of cryptographic hash algorithms during previous course offerings. Since this course has broad coverage of topics in computer security, there is no time to cover cryptography in-depth. The coverage of SHA was aimed to introduce students to the algorithm with the expectation that interested students could independently study the algorithm in more depth. **SHAvisual** was used in class to complement the lecture. Students were also given a take-home problem that required them to work through the operation of the algorithm using **SHAvisual**. After the students had submitted their solutions to the take-home problem, the instructor distributed a survey to the class. Completion of the survey was voluntary.

4. Software Description

SHAvisual is designed to help students learn and instructors teach the SHA-512 algorithm. It supports Windows, MacOS and Linux. **SHAvisual** consists of three major components: the Demo Mode, the Practice Mode and the Full Mode. A separate global view window is also available to highlight the current procedure in the algorithm pipeline. The Demo Mode provides a simplified SHA-512 visualization and is useful for the instructor to demonstrate important operations in the classroom. The Practice Mode is designed for students to learn the detailed computations step by step and perform self-study. A test report system also helps the instructor verify the learning effectiveness. The Full Mode is a full version of the SHA-512 cipher. It takes a plaintext as input and generates the encrypted digest message with major intermediate results shown. Both the Demo Mode and Practice Mode have multiple subpages and the user may access different subpages by clicking their tab names. Buttons are also provided to switch subpages. The Full Mode only uses one subpage to show all the computations. **SHAvisual** always starts with the Demo Mode.

4.1 The Demo Mode

The Demo Mode demonstrates the SHA-512 algorithm step by step. It uses shorter length messages and a single round so that the user can focus on the essential computations rather than repetitive operations. The Demo Mode has five subpages: Message Generation, Workflow Overview, Words Generation, Compression Function and Round Detail.

Message Generation. This subpage demonstrates how to obtain the Augmented Message by expanding the Original Message (plaintext) length to a multiple of 256 (1024 originally) bits (Figure 1). The user clicks the Random Message button to generate a new random plaintext and the corresponding augmented message will be shown at the bottom. Green, blue and red colors indicate the plaintext, padding field and length field, respectively, of the augmented message. Both messages are shown in hexadecimal. Clicking the area of numbers in the Augmented Message portion brings the user to the Workflow Overview subpage.

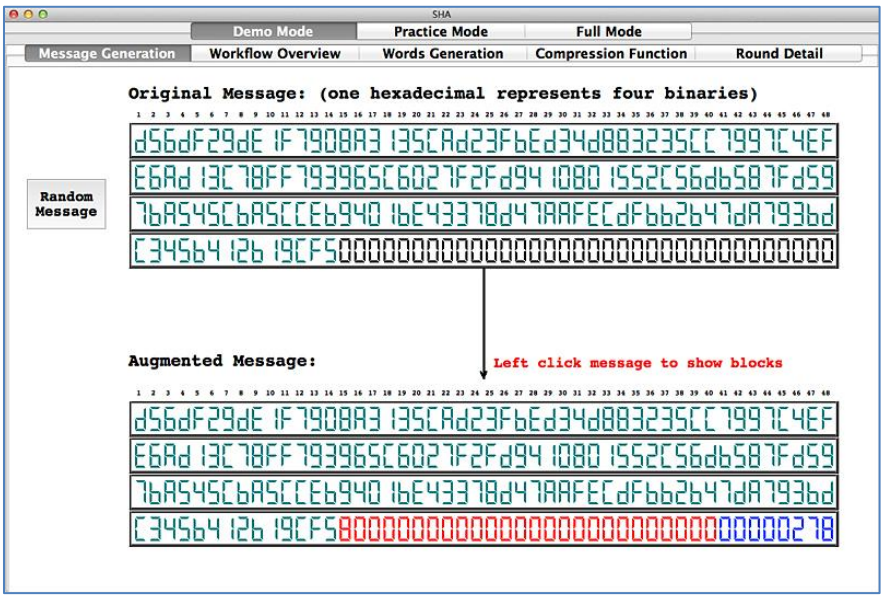

Figure 1: Message Generation of the Demo Mode

Workflow Overview. This subpage offers a general SHA-512 algorithm overview (Figure 2). It illustrates how the final Message Digest is generated using an initial value and the blocks derived from the augmented message. The Initial Value is a 128-bit constant defined by the SHA-512 algorithm and used as an input for the first compression function. Each Block is a 256-bit segment of the augmented message and extended to eighty 16-bit (64-bit originally) words in the Words Generation stage. The user clicks the Block numbers or the Compression Function button to proceed to the Words Generation or Compression Function subpage.

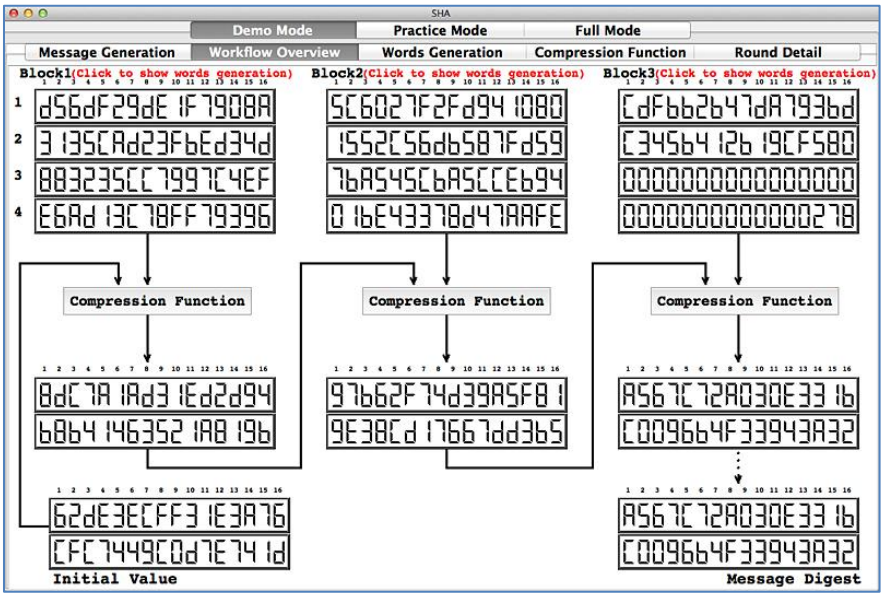

Figure 2: Workflow Overview of the Demo Mode

Words Generation. This subpage demonstrates how the last 64 (in red) out of 80 16-bit (64-bit originally) words are generated from the corresponding block (Figure 3), where the first 16 words (in black) are taken from the block. By sliding the row of words in the upper portion of the page, the user may pick any word (in blue) of the last 64 words to check its generation procedure. Four words used for computation are shown in the middle. The user may click one of the two RotShift buttons to check the detailed operations of the generation (Figure 4). Each word is then used in one round (80 totally) of the corresponding Compression Function.

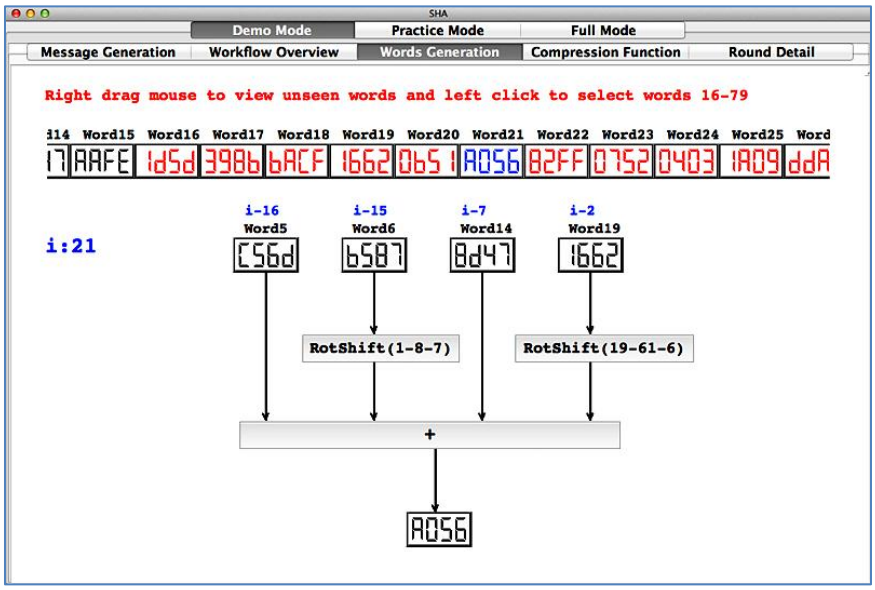

Figure 3: Words Generation the Demo Mode

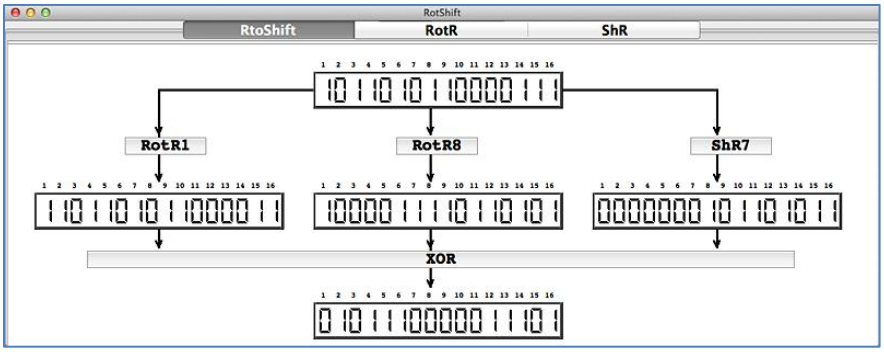

Figure 4: Operations of the Words Generation

Compression Function. This subpage visualizes the pipeline of the Compression Function (Figure 5). The input words A-H are either from the previous compression function or from the initial value in the Workflow Overview (for the first compression function). The output is used for the next compression function or forms the final message digest (for the last compression function). Refer to Figure 2 for the relationship among different compression functions. Word0 is from the Words Generation while Key0 is one of the eighty 16-bit (64-bit originally) constants defined by the algorithm. They will be used in round 0. Eighty similar rounds are needed for the original compression function with one word and one key for each round; but the Demo Mode

only performs one round for clear demonstration. The user clicks the Round0 button to see round details in the Round Detail subpage.

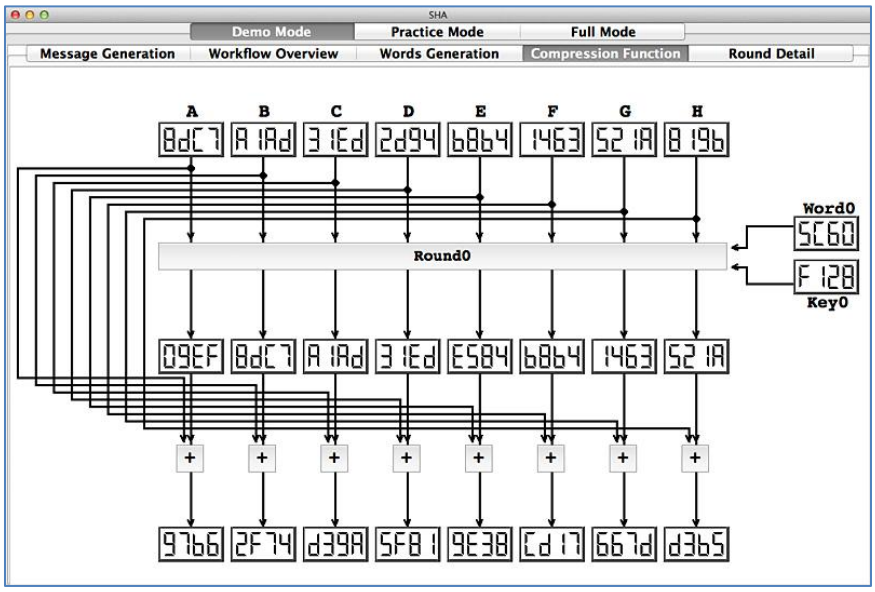

Figure 5: Compression Function of the Demo Mode

Round Detail. This subpage shows the computations of a round. The Round box (Figure 6) in the upper half of this page shows the mapping between the input (A-H) and output of the corresponding round. The lower portion shows the computation of the two new words X and Y in the output. Input A-H, Word0 and Key0 are taken from the corresponding Compression Function. The user clicks the Majority, Rotation and Condition buttons in the Mixer boxes to see the details of computation.

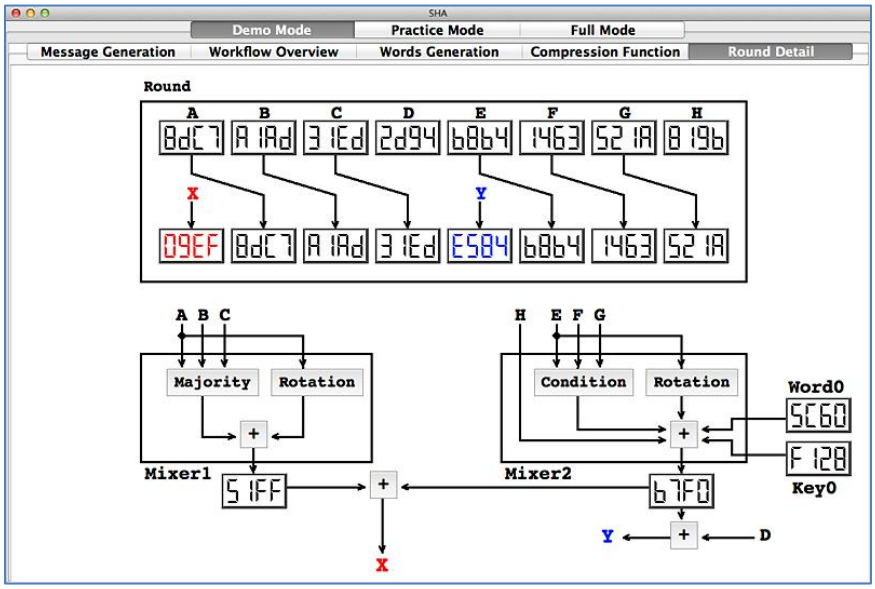

Figure 6: Round Detail of the Demo Mode

4.2 The Practice Mode

The Practice Mode follows the same structure as the Demo Mode (Figure 7) and the user may also step through each computation. However, all results are hidden and a correct answer is required to advance to the next step. The user clicks the Start Practice button to start the process. A dialogue window will pop up to briefly describe the current question. The user should enter an answer and click the Check Ans button. **SHAvisual** will verify the input and display Correct! if the answer is correct, and Wrong! Try it again! otherwise. The user enters a new answer if the current one is wrong. A Show Ans button is provided to show the correct answer and let the user skip the current question. A simple hexadecimal-binary converter is also provided. A Completion Report window (Figure 8) will be shown after the user finishes all questions. The report records the answer to each question using Correct, Wrong or Show Ans according to the user's action. This report may be sent to the instructor to check the student completion rate and evaluate the learning effectiveness.

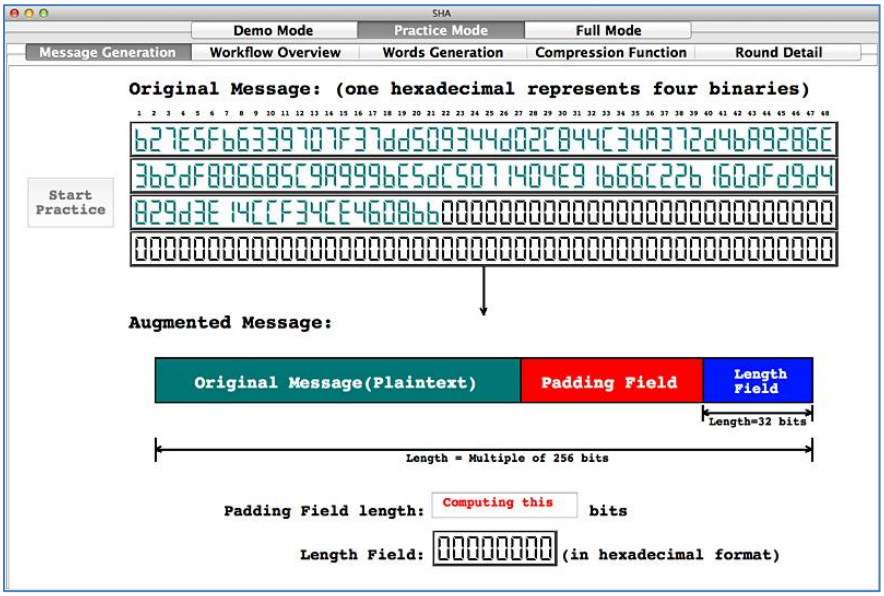

Figure 7: Practice Mode of **SHAvisual**

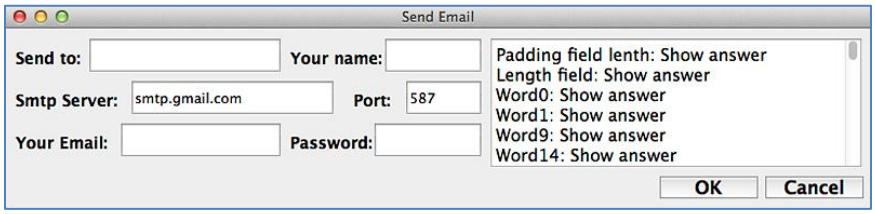

Figure 8: Completion Report of the Practice Mode

4.3 The Full Mode

The Full Mode provides a full version SHA-512 cipher (Figure 9). The user either enters an input string or clicks the Random button to generate a random input. The Clear button allows the user to clear the input and reset the computations. By clicking the Confirm button, the encrypted message digest and important intermediate results are shown.

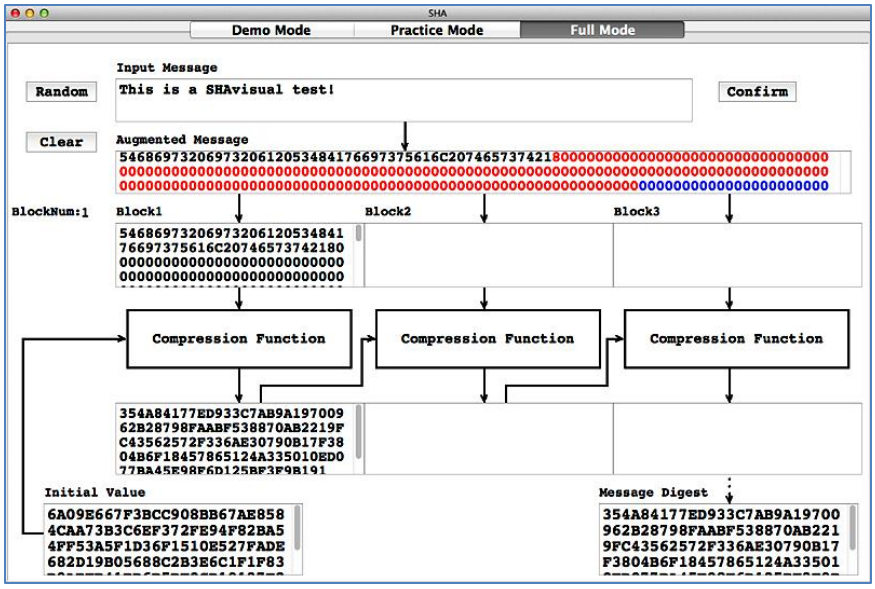

Figure 9: Full Mode of **SHAvisual**

5. Evaluation

The **SHAvisual** survey consists of two major components, twelve questions (Table 1) and eleven write-in comments (Section 5.3). The first nine questions (Q1-Q9) evaluate the effectiveness of **SHAvisual** (EEQ) and the other three (Q10-Q12) investigate the use of **SHAvisual** (UIQ). The EEQ questions have the same set of choices: 1: "Strongly Disagree", 2: "Disagree", 3: "Neutral", 4: "Agree", and 5: "Strongly Agree". The choices for Q10 are 1:less than 5 mins, 2:5-10 mins, 3:10-15 mins, 4:15-30 mins, 5:over 30 mins. The choices for Q11 are 1:only once, 2:1-3 times, 3:3-5 times, 4:5-10 times, 5:over 10 times. The choices for Q12 are 1:less than 5 mins, 2:5-15 mins, 3:15-30 mins, 4:30-60 mins, 5:over 1 hour. We collected 24 valid survey forms from two disciplines: 19 in computer science and software engineering (CS) and five in electrical and computer engineering (ECE).

Table 1: Survey Questions

ID	Question
Q1	Demo mode helped me better understand the work flow of the SHA cipher
Q2	Demo mode was helpful for my self-study
Q3	Practice mode helped me remember SHA encryption
Q4	Full mode helped me understand how the SHA cipher encrypts a full-length message
Q5	Full mode provided me a simple tool to do SHA encryption
Q6	Global view helped me locate the current demonstrated operation
Q7	Using **SHAvisual** I was able to identify the parts of the SHA cipher that I did not understand before
Q8	I was able to better understand the SHA algorithm with **SHAvisual**
Q9	The SHA software enhanced the course
Q10	How long did it take to understand SHA Algorithm with **SHAvisual**
Q11	How often did you use **SHAvisual**
Q12	How long did you use **SHAvisual**

5.1 General Discussion

Table 2 shows the mean (μ), standard deviation (σ) and confidence interval (CI-,CI+) at 95% confidence level and Figure 10 has the interval plot of Table 2. For EEQ (effectiveness evaluation questions), the highest score of 4.2 was given to Q8, indicating that students highly agreed that **SHAvisual** helped them understand the SHA algorithm better. Q1, Q3 and Q7 all received the same high score of 4.0, suggesting that both the Demo and Practice modes had a positive impact on student learning. Except for Q6, other questions were rated in the range from 3.5 to 3.9, still above the neutral rating 3.0. The lowest score of 3.3 was given to Q6, indicating that the global view slightly helped students identify the relationship between the current operation and the overall algorithm. Hence, although the rating of EEQ varied among questions with standard deviations in a small range from 0.7 to 0.9, the general trend was still positive.

Table 2: Mean μ, Standard Deviation σ, and Confidence Interval

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
μ	4.04	3.83	4.04	3.50	3.58	3.29	4.0	4.17	3.92	3.67	2.33	4.58
σ	0.81	0.82	0.86	0.78	0.78	0.75	0.78	0.76	0.65	1.13	1.09	0.65
CI-	3.72	3.51	3.70	3.19	3.27	2.99	3.69	3.86	3.66	3.21	1.90	4.32
CI+	4.36	4.16	4.39	3.81	3.89	3.59	4.31	4.47	4.18	4.12	2.77	4.84

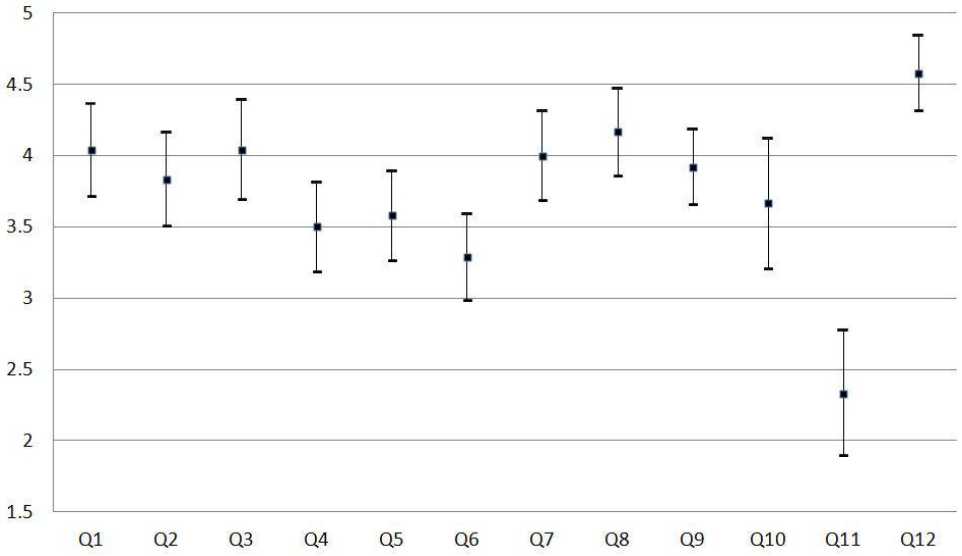

Figure 10: Interval Plot of Table 2

For UIQ (usage investigation questions), Q12 had a very high average (4.6), indicating that students used the software for nearly an hour. The average of Q10 was 3.7, which means that the majority of the students took less than 15 minutes to understand the SHA algorithm. However, most students only used the tool a few times as indicated by the average 2.3 of Q11. Note that the standard deviations of Q10 and Q11 were slightly larger than 1.1. Table 3 has the distributions of answers. For Q10, around 33% of all students took 10 to 15 minutes to understand the algorithm with the tool while another 33% took more than 30 minutes. The distribution of Q11 indicated that more than 90% of all students used the tool less than five times.

Q12 showed that more than two-third of the students (67%) used the tool for over an hour while a quarter of them used it 30 minutes to one hour. The Spearman rank test was used to determine if the correlations among UIQ question pairs are significant. The null hypothesis is that there is no correlation under the level of significance $\alpha = 0.05$. The test results showed that only Q10 and Q12 had a significant positive correlation with the *p*-value being 0.002. Hence, students who took more time to understand the SHA algorithm with the tool may also have spent a longer time using the tool.

Table 3: Usage Answer Distributions

	Choice 1	Choice 2	Choice 3	Choice 4	Choice 5
Q10	0.00	0.17	0.33	0.17	0.33
Q11	0.21	0.42	0.29	0.00	0.08
Q12	0.00	0.00	0.08	0.25	0.67

5.2 Further Statistical Analysis

We also investigated the possible relation between questions Q1-Q9 (student reactions) and Q10-Q12 (the use of **SHAvisual**). Student reactions were divided into two groups based on Table 4 for questions Q10 to Q12. The null hypothesis for this study was: the time spent on understanding the SHA algorithm (Q10), the number of times using this tool (Q11), and the total time spent on this tool (Q12) have no impact on student reactions on the EEQ questions. The level of significance is $\alpha = 0.05$.

Table 4: Student Reactions Grouping

	Group 1	Group 2
Q10	≤ 15 mins (12, 50%)	> 15 mins (12, 50%)
Q11	≤ 3 times (15, 62%)	> 3 times (9, 38%)
Q12	≤ 1 hour (8, 33%)	> 1 hour (16, 67%)

Table 5 shows the *p*-values of our ANOVA (**AN**alysis **O**f **VA**riance) study. The two smallest *p*-values were 0.054 (between Q6 groups with respect to Q10) and 0.006 (between Q5 groups with respect to Q12). Since all other *p*-values were larger than the chosen $\alpha = 0.05$, the null hypothesis cannot be rejected. Hence, we have strong evidence showing that the student reactions, except for the indicated two cases, were not affected by the time spent on understanding the SHA algorithm (Q10), the number of times they used the tool (Q11), and the time spent on using the tool (Q12).

Table 5: ANOVA Results for Three Groupings

Grouping Question	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
Q10	0.807	0.628	0.488	0.611	1.000	0.054	1.000	0.603	0.219
Q11	0.485	0.803	0.859	0.794	0.353	0.839	0.600	0.789	0.639
Q12	0.676	0.864	0.871	0.097	0.006	0.348	0.096	0.191	0.831

Q6 asked students if the global view helped them locate the currently demonstrated operation. The *p*-value of Q10 between the group taking no more than 15 minutes to understand the SHA

algorithm ($\mu = 3.6$) and the other group ($\mu = 3.0$) was 0.054, which is barely larger than the chosen level of significance $\alpha = 0.05$. Although the means of these groups showed a 0.6 difference, statistically we could not reject the null hypothesis. Q5 asked students if the Full mode offered a simple tool to perform SHA encryption. Grouping based on Q12 showed a *p*-value of 0.006, which is smaller than the chosen level of significance $\alpha = 0.05$ and, hence, the null hypothesis is rejected. This means that the group of students using the tool for more than one hour ($\mu = 3.9$) had a significantly different response to Q12 from the group of students using the tool for no more than one hour ($\mu = 3.0$).

This ANOVA analysis treated each question individually. To address the possible dependence among the nine EEQ questions, we also conducted a MANONA (Multivariate ANOVA) analysis by considering all nine questions simultaneously. Wilk's lambda test suggested that we cannot reject the null hypothesis (i.e., no group differences) under the chosen level of significance $\alpha = 0.05$. Based on these findings, we have sufficient evidence to conclude that student reactions are generally independent of the time they spent on understanding the SHA algorithm, and the time (and the number of times) they spent on using the software.

5.3 Student Comments

The set of 11 write-in questions was designed to allow students to make suggestions for future improvement. We focused on the following topics: whether the restrictions of the small-size input message and the single round demonstration affected student learning, whether the Words Generation module is useful, whether the Compression Function module needs improvement, whether the Round/Mixer module is good enough, the evaluation of Demo, Practice and Full modes, and software installation issues.

Student comments showed that the restrictions of the small-size message and single round demonstration did not affect their learning of the SHA algorithm. Students said "*It gave good insight to blocks*", "*A larger message size would have made it more confusing*", "*Concepts are the same even if the size is small*", and "*It was focused and made the inner workings of the round clearer*". One student suggested that defining a minimum message size would be useful while another student thought that a second round may be helpful although the idea was already clear after the first round.

The feedback on the Words Generation module was almost completely positive with comments like: "*Word pattern was pretty well shown, which enhanced my learning*" and "*The fact that it was visually laid out was helpful*". We received some suggestions, such as using a bubble dialog to show a detailed explanation. The comments on the Compression Function and Round/Mixer modules followed the same trend. The majority of students felt these modules were helpful by pointing out that "*The mode was illustrative*" and "*That would have been extremely confusing as just a formula and the visual aspect helped*". One student mentioned that the Compression Function module was a little confusing due to a "deep nesting" structure; however, this student also agreed that this was the nature of the algorithm.

Students agreed that the Practice mode was effective and indicated "*I liked that you are able to step through the process*", "*Clearly marked + Straightforward*", "*Great way to learn*", "*Well*

laid out and easy to follow", "*Most effective component*", "*Useful as an application of principles*", and "*It helped me understand SHA*". Some improvements were mentioned, such as adding a hint button to provide a brief reminder and adding more connections between input and output windows.

The Full mode received fewer comments since most students did not use it. This is not surprising because **SHAvisual** was released right before the evaluation and students were required to use the Demo and Practice modes as homework. Thus, time for the students to have a more comprehensive use of the tool was limited. We saw some positive comments, for example: "*It was neat to be able to use the 'real deal'*".

Compared with blackboard work, almost all students agreed that the Demo mode was more useful for them to learn the SHA algorithm. Typical comments were "*I was able to use it on my own later to reinforce what I learned in class*", "*I like the more dynamic nature*", "*It definitely helped following a program rather than using the blackboard. Visualization would make it even better*", "*It was easier to see how items connected with each other*", "*The best part was that it really enforced where the data came from and what was done to it*", and "*The structure of the algorithm is very nested and the demo gave a good overview and let you see details of each piece*". Thus, we believe that **SHAvisual** provided students with an effective way to learn the algorithm.

Students also gave some general comments for improvement. For example, they suggested adding a help button to explain each step, integrating the Dec-Hex conversion into the current converter, and giving a brief explanation when the answer is wrong in the Practice mode. Two students reported installation issues on MacOS as they perhaps failed to install the needed libraries properly. Windows and Linux users did not have major installation problems.

In summary, we believe that **SHAvisual** has fulfilled its purpose, helping students learn and the instructors teach the SHA algorithm effectively. With the suggestions, we should be able to improve **SHAvisual** significantly in the near future.

5.4 Self-Study Investigation

We also invited students who did not take our course for a two-stage self-study. This small scale survey was used to determine if there was a difference between classroom and self-study with our tool. There were two stages; each stage took about one week. In Stage 1, volunteers were asked to find resources (e.g., books, articles, Internet sites, etc.) to learn the SHA algorithm. At the end of Stage 1, volunteers evaluated their progress. In Stage 2, volunteers were provided with **SHAvisual** and a brief user guide. At the end of this stage, volunteers filled out the evaluation form, which was identical to the one used in classroom evaluation.

We collected six completed survey forms. Table 6 has the mean (μ), standard deviation (σ), and confidence interval (CI-, CI+) at 95% confidence level. Figure 11 shows the interval plot of Table 6 and Figure 12 is a comparison of the means from our class survey (Table 2) and self-study (Table 6). Since we would like to know if the classroom evaluation and self-study evaluation had any differences, statistical hypothesis tests were applied to each question pair.

The tests used were Student's *t*-test and the Mann-Whitney *U*-test. Note that Student's *t*-test compares the sample means, while the Mann-Whitney *U*-test compares the sample medians. Moreover, the Mann-Whitney *U*-test does not require the involved populations to be normally distributed. The null hypothesis for each corresponding question pair was that the mean (resp., median) obtained from the classroom evaluation and the mean (resp., median) obtained from the self-study were the same. The *t*-test and *U*-test rows in Table 6 are the *p*-values obtained from the corresponding *t*-test and *U*-test, respectively. Both tests showed that, except for Q3, Q5 and Q6, ratings by students in our class and ratings by participants in our self-study did not have a statistically significant difference. Since Q3's rating from self-study and classroom evaluations were 3.17 and 4.04, respectively, students taking our class felt that the Practice mode is more helpful than participants in our self-study. On the other hand, participants in our self-study rated Q5 and Q6 (4.33 and 4.00) higher than those in our class (3.58 and 3.29).

Table 6: Mean μ, Standard Deviation σ, and Confidence Interval (Self-Study)

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
μ	4.33	4.17	3.17	3.50	4.33	4.00	4.00	4.00	3.83	4.00	2.17	2.17
σ	0.52	0.75	0.75	0.84	0.52	0.63	0.89	0.89	0.98	0.89	1.33	0.75
CI-	3.92	3.56	2.56	2.83	3.92	3.49	3.28	3.28	3.05	3.28	1.10	3.56
CI+	4.75	4.77	3.77	4.17	4.75	4.51	4.72	4.72	4.62	4.72	3.23	4.77
t-test	0.24	0.32	0.049	1.00	0.02	0.047	1	0.63	0.88	0.50	0.83	0.24
U-test	0.45	0.39	0.035	0.84	0.03	0.043	0.98	0.63	0.93	0.53	0.79	0.16

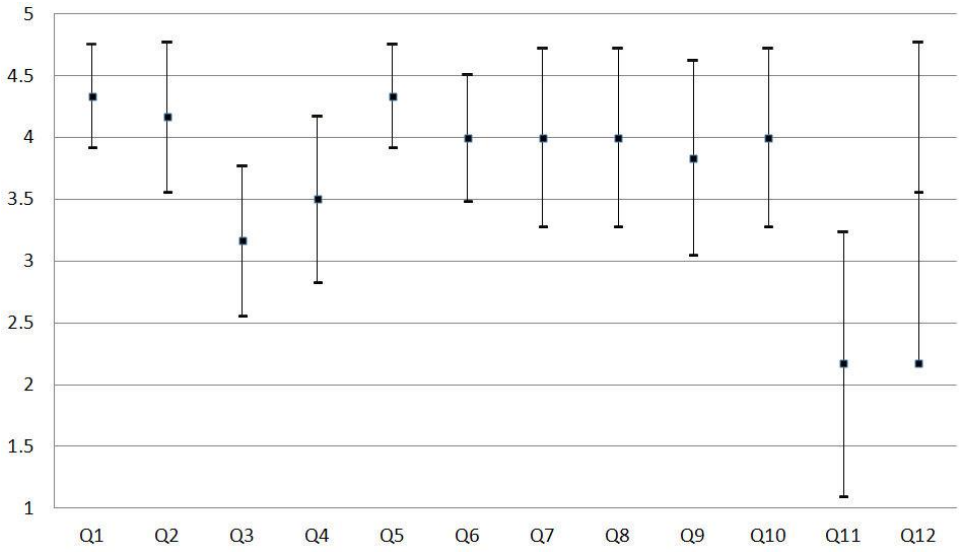

Figure 11: Interval Plot of Table 6

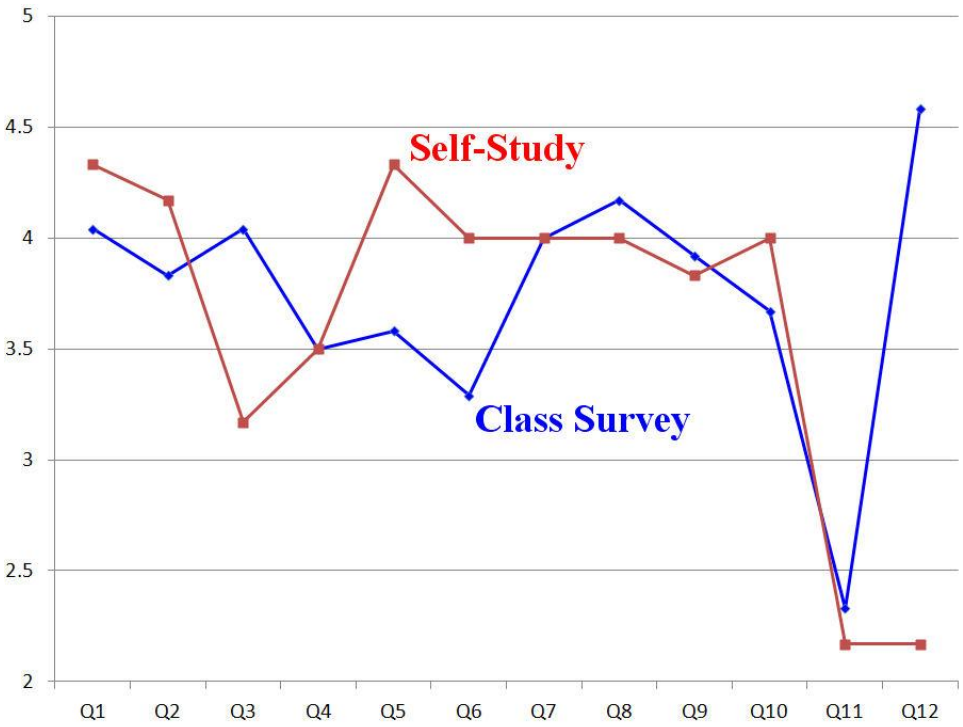

Figure 12: Class Survey vs. Self-Study

At the end of each stage, volunteers did an exercise. The problems on both exercises were similar and addressed the key components of the SHA algorithm. However, the second exercise had more problems asking for some details of the SHA algorithm. We were able to extract a few interesting findings. First, every volunteer finished the questions on the Compression function with the averages (normalized to 100% for comparison purposes) of exercise 1 and exercise 2 being 90% and 96.7%. Both the pair *t*-test and the Mann-Whitney *U*-test indicated that the difference was not significant with *p*-values being 0.347 and 0.374, respectively. Second, there were two problems in exercise 1 and four problems in exercise 2 which addressed the remaining non-padding related issues. Some volunteers did not finish the problems in exercise 2. However, a *t*-test indicated that the mean of exercise 1 (100%) and the mean of exercise 2 (87.5%) did not have a significant difference ($p = 0.5$). Third, volunteers did not perform well on the message padding and word generation problems with means 45.83% and 33.33% for exercise 1 and exercise 2, respectively. Again, a *t*-test did not suggest a significant difference ($p = 0.538$). The results suggest that students' performance did not significantly change after having access to the software. However, we do not feel this is a good indicator of the students' understanding. Because this survey was conducted near the end of a semester, students were working hard to finish their own course work and prepare for their final exams. We found from the returned forms that volunteers had the tendency to skip some problems that required lengthy calculations, especially those problems in the second exercise that require hexadecimal arithmetic. As a result, a direct comparison of these two exercises became difficult.

5.5 Instructor Experience

It is obvious that **SHAvisual** allows the details of the algorithm to be demonstrated over relatively large inputs without using the blackboard and without the possibility of small

mathematical errors that might confuse students. In addition, **SHAvisual** also permits an instructor better control over how much class time is devoted to covering the algorithm, because the tool is suitable for self-study. On the other hand, from our early work **ThreadMentor** for visualizing thread execution behavior[6] to our cryptography visualization tools, a handful of students indicated that they would be in favor of a non-visual environment. These students preferred an algorithmic presentation rather than a visual-based presentation. Moreover, some students preferred blackboard work because they believed that the use of visuals caused less classroom interaction and participation. In this regard, the instructor may have to carefully mix the blackboard work and the use of **SHAvisual** in a way that is suitable to the students. Given our experience, fully relying on the tool could disengage the possible interaction among the instructor, the students, and the real ideas and perhaps the deep concepts of the algorithm. In general, the Demo mode has to be paused at critical points for the instructor to inject and explain the key ingredients. Otherwise, the visual component would just become an automated animation that defeats the original design principle of **SHAvisual**.

As mentioned at the end of Section 5.4, volunteers did not perform well in the Message Padding and Word Generation components. These are rather complicated steps. The instructor may have to pay more attention to explain the concepts so that the students could follow the "flow" of these two components. Because **SHAvisual** is able to provide an essentially infinite number of examples for students to work through with no additional effort by the instructor, the instructor may ask the students to play the Demo mode extensively and use the Practice mode to actually do a few more exercises. **SHAvisual** has a reporting system with which the activities when a student uses the Practice mode to solve problems will be reported to the instructor for evaluating learning effectiveness. While this reporting system is rudimentary, it does provide a way for the students to gauge their progress. Thus, the instructor could encourage the students to use the Demo and Practice modes extensively and perhaps to switch between these two modes for verifying the concepts and operations while solving a problem. Some instructors at our SIGCSE 2015 workshop[5] raised the concern that it is difficult to give homework over this material. The authors will investigate a more full-featured reporting system with better quiz functionality in the near future.

6. Conclusions

This paper presented a visualization tool **SHAvisual** for teaching and learning the SHA-512 algorithm. With the tool, instructors are able to present all details and inner working of different components of the algorithm. It also helps students see the flow of the algorithm, learn the concepts and practice the computation steps using the Practice mode. Evaluation results showed that **SHAvisual** was effective in the classroom presentation and for student self-study. The grouping analysis indicated that the time spent on understanding the algorithm, the number of times using the tool, and the total time spent on the tool did not impact the student evaluation results.

SHAvisual uses shorter length messages and a single round in the Demo and Practice modes. While nearly all students indicated these restrictions worked fine, we are currently looking at ways to extend **SHAvisual** in order to alleviate such limitations and improve the tool in the near future. Based on the student comments, the most needed extensions are (1) supporting the

selection of multiple message lengths, (2) adding one more round to illustrate the connection between two consecutive rounds of a compression function, (3) providing help buttons for each subpage, and (4) considering a better visualization style for the nesting algorithm structure in the compression function subpage (e.g., better color schemes).

SHAvisual is a part of larger development of cryptography visualization tools supported by the National Science Foundation. In addition to **SHAvisual**, **VIGvisual**[9] for the Vigenère cipher, **DESvisual**[15] for the DES cipher, **ECvisual**[16] for the elliptic curve based ciphers and **RSAvisual**[17] for RSA cipher are available online and new visualization tools for AES cipher (e.g., **AESvisual**) will become available soon. Tools, evaluation forms, and installation and user guides for Linux, MacOS and Windows can be found at the following URL:

www.cs.mtu.edu/~shene/NSF-4.

Acknowledgments

The authors are supported by the National Science Foundation under grants DUE-1140512, DUE-1245310 and IIS-1456763.

References

1. *Cryptography Demos*. http://nsfsecurity.pr.erau.edu/crypto/index.html

2. *Cryptool*. http://www.cryptool.org

3. M. S. Asseisah and H. M. Bahig, Visual Exploration of Classical Encryption on the Web, *The Ninth IASTED International Conference on Web-based Education (March 15-17, 2010)*, 2010.

4. M. S. Asseisah, H. M. Bahig, and S. S. Daoud, Interactive Visualization System for DES, *Proceedings of the 6th International Conference on Active Media Technology*, pages 18-25. 2010.

5. S. Carr, M. Keranen, and J. Mayo, Teaching Cryptography and Access Control Hands-On (Abstract Only), *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pages 707-707, 2015.

6. S. Carr, J. Mayo, and C.-K. Shene, **ThreadMentor**: A Pedagogical Tool for Multithreaded Programming, *Journal on Educational Resources in Computing*, 3(1), 2003.

7. G. Cattaneo, A. D. Santis, and U. F. Petrillo, Visualization of Cryptographic Protocols with GRACE, *Journal of Visual Languages & Computing*, 19(2):258-290, 2008.

8. D. E. R. Denning, *Cryptography and Data Security*, Addison-Wesley, 1982.

9. C. Li, J. Ma, J. Tao, C.-K. Shene, M. Melissa and C. Wang, **VIGvisual**: A Visualization Tool for the Vigenère cipher, to appear in *Proceedings of the 20th annual Conference on Innovation and technology in Computer Science Education*, 2015.

10. A. J. Menezes, P. C. V. Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

11. D. B. Nasr, H. M. Bahig, and S. S. Daoud, Visualizing Secure Hash Algorithm (SHA-1) on the Web, *Proceedings of the 7th International Conference on Active Media Technology*, pages 101-112. 2011.

12. A. Salomaa, *Public-Key Cryptography*, Springer-Verlag, 1992.

13. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, John Wiley, 1995.

14. D. Schweitzer and L. Baird, The Design and Use of Interactive Visualization Applets for Teaching Ciphers, *IEEE Information Assurance Workshop*, pages 69-75, 2006.

15. J. Tao, J. Ma, M. Keranen, J. Mayo, and C.-K. Shene, **DESvisual**: A Visualization Tool for the DES Cipher, *Journal of Computing Sciences in Colleges*, 27(1):81-89, 2011.

16. J. Tao, J. Ma, M. Keranen, J. Mayo, and C.-K. Shene, **ECvisual**: A Visualization Tool for Elliptic Curve Based Ciphers, *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, pages 571-576, 2012.

17. J. Tao, J. Ma, M. Keranen, C.-K. Shene, and C. Wang, **RSAvisual**: A Visualization Tool for the RSA Cipher, *Proceedings of the 44rd ACM Technical Symposium on Computer Science Education*, pages 635-640, 2014.