DESvisual: A Visualization Tool for the DES Cipher

Jun Tao, Jun Ma, Melissa Keranen, Jean Mayo, Ching-Kuang Shene*
Department of Computer Science
Michigan Technological University
Houghton, MI 49931
Email: {junt, junm, msjukuri, jmayo, shene}@mtu.edu

March 26, 2011

Abstract

This paper describes a visualization tool DESvisual that helps students understand and instructors teach the building blocks of symmetric encryption. In particular, the tool depicts the primitive operations required to perform the initial permutation and one Feistel round of DES using either an eight or 16 bit input. A student can trace through an encryption performed by the tool, or can be guided through an encryption or decryption, computing the output of each operation herself. This helps students to understand the primitive operations, how these operations are composed into the DES algorithm, and how functions and their composition are depicted and documented. Furthermore, the opportunity for self-study provides an instructor greater flexibility in selecting a lecture pace over this detail-filled material.

Security concerns impact an ever increasing number of the applications that computer scientists design and develop. Many security problems are solved through use of cryptography. Unfortunately, computer science students commonly have difficulty with the sophisticated mathematics used in cryptographic algorithms. This problem is exacerbated by the fact

*Communicating Author

that cryptography commonly competes for attention with many other important topics in an undergraduate curriculum.

Well-designed pedagogical tools can greatly help reduce the difficulty in teaching and learning complex topics. However, tools for teaching and learning cryptography is at best minimal. To the best of our knowledge, only two papers published in recent years are mainly dedicated to cryptography pedagogy [3, 4]. One of them [4] discusses two simple algorithms, DES and AES, using spreadsheets, while the other [3] uses computer algebra systems such as Maple. There are books which include C, C++ or Java source code [1, 5, 6, 9]. However, only [1] is a textbook that uses Java applets as examples, and its algorithm-example-sample program approach may not be the best way for many instructors and students.

In this paper, we describe a visualization tool DESvisual that helps students understand the building blocks of symmetric encryption. In particular, DESvisual depicts the primitive operations required to perform the initial permutation and one Feistel round of DES using an 8- or 16-bit input. A student can trace through an encryption performed by the tool, or can be guided through an encryption or decryption, computing the output of each operation herself. This helps students understand the primitive operations, how these operations are composed into the DES algorithm, and how functions and their composition are depicted and documented. The opportunity for self-study provides an instructor greater flexibility in selecting a lecture pace over this detail-filled material. DESvisual is part of a larger project to provide visualization tools to help address the challenges of teaching cryptography.

DESvisual was used in a junior-level introduction to cryptography course. The evaluation showed that DESvisual is effective; it can help students learn and practice effectively and efficiently and help instructors to easily present the DES algorithm.

In the following, Section 1 provides the background of our cryptography course, Section 2 presents our visualization tool, Section 3 discusses use of DESvisual in the classroom, Section 4 has a detailed study of our findings from a survey, and Section 5 is our conclusion.

1 Course Information

DESvisual was used in a cryptography course, MA3203 Introduction to Cryptography, that was offered by the Department of Mathematical Sciences at Michigan Technological University. This is a junior level course that gives a basic introduction to the field of Cryptography.

Our course covers classical cryptography, the Data Encryption Standard (DES), the Advanced Encryption Standard (AES), the RSA algorithm, discrete logarithms, hash functions, and elliptic curve cryptography. For each cryptosystem, we study how it was designed, why it works, how one may attack the system, and how it has been used in practice.

DES was the official data encryption standard from 1977 to 2000 and has been an important part of the field of cryptography. Therefore, our course gives a good amount of attention to this algorithm. Even though each step of the algorithm is straightforward, students at times find it difficult to put all of the pieces together. When asked to go through one entire iteration of the algorithm, they often find they do not understand the algorithm in its entirety. One needs to give a thorough treatment of DES when teaching cryptography. One reason to do so is because DES plays an important part in the history of cryptography. Another reason is because understanding DES helps one to understand the Advanced Encryption Standard. If students have a good grasp on DES, then teaching them AES becomes routine because the methods used in the AES algorithm are similar to that of DES.

2 DESvisual Description

DES encryption is comprised of an initial permutation (IP), sixteen Feistel rounds, and a final permutation. DESvisual computes an IP and one Feistel round. Figure 1 depicts the Main window containing the IP and Feistel computation. The Overview window of Figure 2 appears when the What's This button is clicked from the Main window; it shows the relationship between the tool computations and a full DES encryption. Computations are performed on either 8 or 16 bit inputs and 6 or 10 bit keys (Use 16 bits in the Main window).

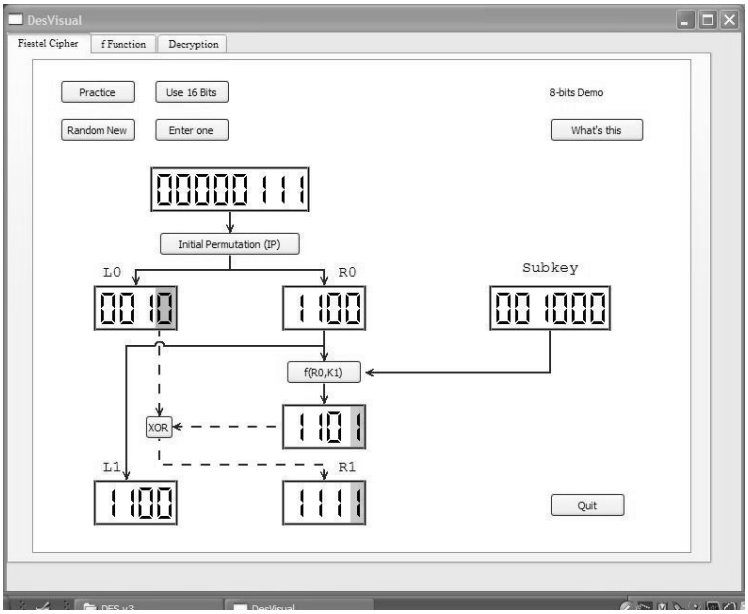

Figure 1: Main Window - IP & Feistel Cipher

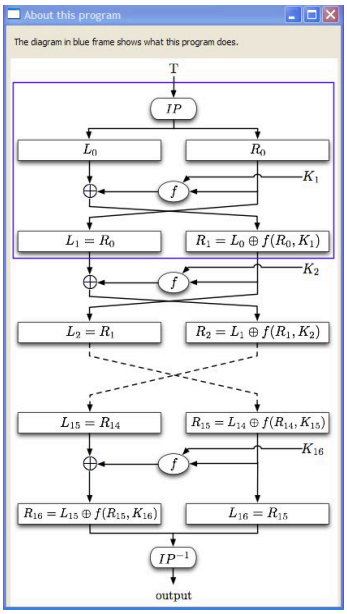

Figure 2: Overview Window

A user may either have the system generate a random input and subkey (Random New) or enter their own input and subkey (Enter one).

There are three modes of operation. A user can (1) trace an encryption performed by the tool, (2) perform a guided encryption on the selected input, or (3) perform a guided decryption on the output from an encryption. In a guided encryption (or decryption), the student is prompted to perform the computation for each operation of the encryption (or decryption) in turn. These modes are described below.

Trace A user can trace an encryption (or decryption) by tracking a specific bit across each operation. In this mode, the tool performs all computations. Clicking on a bit traces that bit across the operation from which it was derived. The dashed arrows and highlighted bits of Figure 1 depict a bit trace. (These bits and arrows are highlighted in red in the tool.) The f function is traced in a separate window that appears by pressing the f(R0,K1) button from the Main window. The f function window is shown in Figure 3. Pressing the Initial Permutation or Expand buttons opens a window that contains the corresponding table. Depressing the S-box button causes the corresponding table to be displayed and the

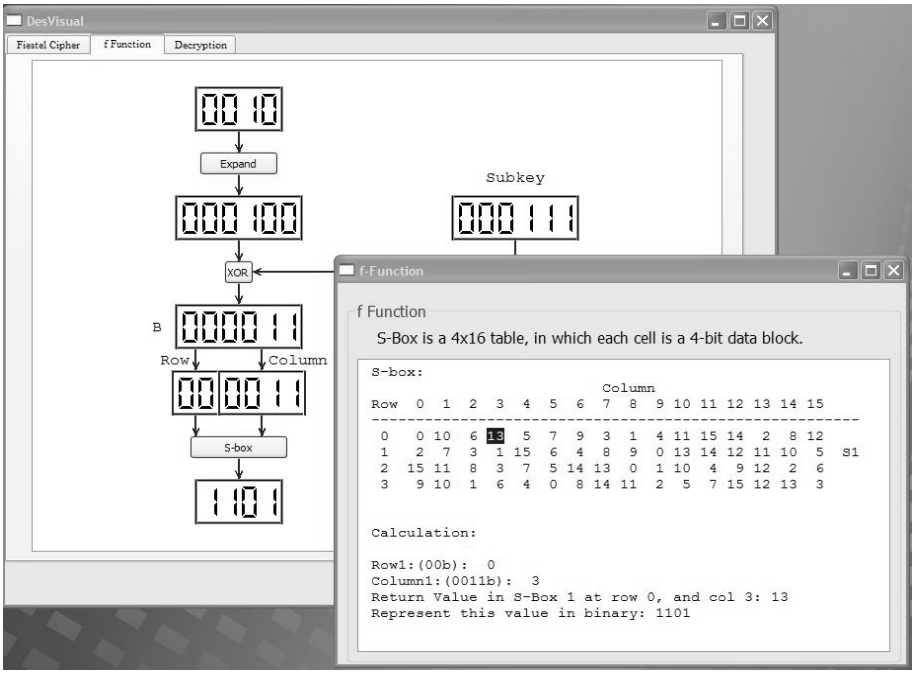

Figure 3: F Function and S-box Table

output element is highlighted. This is depicted in Figure 3.

Guided Encryption and Decryption In each of these guided modes, the tool steps through each operation of the cipher and asks the student to compute the output from the current operation. A guided encryption begins when the Practice button is depressed from the Main window. A decryption begins by selecting the Decryption tab from the Main window. One step of a guided decryption is depicted in Figure 4. Input to the decryption is the output from the encryption calculation.

3 Classroom Use

DESvisual was used in an introductory cryptography course (Section 1). The course is geared towards mathematics majors, but typically the majority of students are not math majors, and many of them are from computer science related fields.

The demonstration mode cf DESvisual was used for classroom presentation. The instructor described each part of the DES algorithm in detail using the blackboard in typical lecture

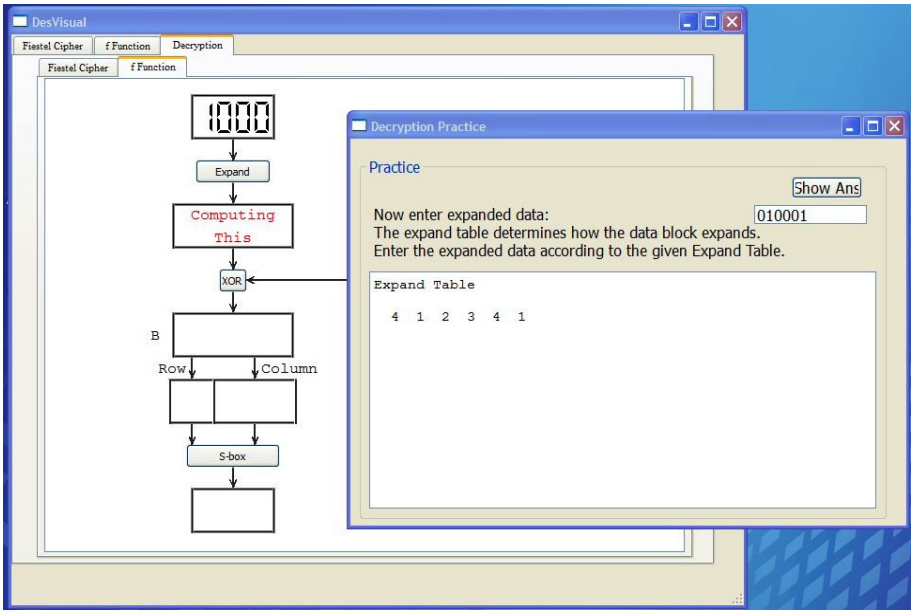

Figure 4: Guided Decryption

style. In the next class period, the instructor demonstrated the algorithm to the class by using the software. The software allowed the instructor to demonstrate one entire round of the cipher very quickly and completely. The students were able to see how each step of the algorithm fits together. By using the software in the second class period, the instructor was able to summarize everything that was previously taught in detail. This allowed students to see the big picture.

After the demonstration of the software to the class, three versions of DESvisual (Linux, MacOS and Windows) were was made available to students to download on their own computer. At this point, the software was mostly used in practice mode. In practice mode, students can enter a plaintext message into the system, and then encrypt it step by step on their own. The program will tell them at each step if they are correct. Students often find that when they try encrypting a plaintext on their own, they then identify the parts of the algorithm which are unclear to them. They were able to use the software at home while learning about differential cryptanalysis in class. Differential cryptanalysis is a method that can be used to attack DES. In order for one to understand this process, they must understand

the encryption method used in DES.

After the students had access to DESvisual for a week, the instructor performed a survey in class. Extra credit was offered to anyone who used the software and completed the survey.

4 Evaluation

The survey forms showed that there are 17 students in computer science, 6 in mathematics, and 11 in other disciplines. Of these 11 students, 4 are computer network and system administration majors, 5 are computer and electric engineering majors, 1 is in material engineering, and 1 is unknown. Because we intend to study the impact of DESvisual on each discipline, we asked students to fill in their disciplines instead of using a class roster.

The DESvisual survey consists of two components, a set of six questions and 12 write-in comments. The six questions are listed in Table 1. Choices available are 5:strongly agree, 4:agree, 3:neutral, 2:disagree, and 1:strongly disagree. Although there were 38 students in this class, we had only 34 valid samples. Table 2 summaries our findings. Since the Other row has only two samples, it will not be discussed. In general, reactions to DESvisual are positive. The highest score is given to Question 1 (DESvisual being able to reveal the inner working of the DES algorithm), followed by Question 2 (helping students understand the DES cipher more). The lowest score is that of Question 4, suggesting that the demo mode is less helpful than the practice mode.

Table 1: Survey Questions

Numberr	Question
1	DESvisual helped see the inner workings of the DES cipher
2	I understood the DES cipher more after I was able to use DESvisual
3	With DESvisual I was able to identify the parts of the algorithm that I did not understand
4	The demo mode was helpful for my self-study
5	The practice mode helped me understand the DES operations
6	DESvisual enhanced the course

Table 2: Survey Summary
CNSA: Computer Network and System Administration, CpE: Computer and Electric Engineering
CS: Computer Science, Math: Mathematics
Numbers are presented as mean(variance)

Major	Samples	1	2	3	4	5	6
CNSA	4	3.3(0.9)	2.6(1.6)	3.8(1.6)	3.3(0.9)	3.0(2.0)	3.0(2.0)
CpE	5	3.4(1.3)	3.2(1.7)	3.2(1.7)	3.4(0.8)	3.2(1.7)	3.2(1.7)
CS	17	3.6(0.8)	3.8(0.8)	3.4(1.4)	3.2(1.3)	3.3(1.5)	3.5(1.1)
Math	6	3.5(1.1)	2.8(0.6)	3.0(0.4)	2.8(1.4)	3.3(1.5)	3.2(1.0)
Other	2	4.5(0.5)	4.5(0.5)	4.0(2.0)	3.5(0.5)	4.5(0.6)	4.0(0.0)
Overall	34	3.6(0.9)	3.5(1.1)	3.4(1.2)	3.2(1.1)	3.3(1.4)	3.4(1.3)

Because this class has students from five disciplines, it is very helpful to understand the differences among these groups. Table 2 shows some interesting results. First, CNSA majors gave the lowest score to Question 2 and the highest to Question 3; but, the variances in general are larger than those of other majors, except for CpE. This suggests that student responses vary significantly. CpE majors also had large variances with the exception for Question 4. In other words, CpE students uniformly considered the demo mode of DESvisual being helpful for self-study; but, variances were large for other issues. Second, CS and Math majors had similar variances with the exception of Question 3. Math majors were neutral about identifying the parts of the DES algorithm with DESvisual while CS majors were somewhat split on this issue due to a high variance. This may be because CS majors are more algorithmic and more sensitive to the building blocks of an algorithm. Furthermore, some of them may not need a visualization component to understand the DES algorithm. Third, except for Question 4, CS majors consistently scored DESvisual higher than other majors with smaller variances.

Table 3 is the correlation matrix of the scores in Table 2. It is clear that the correlation between any two majors is rather low except for the one between CNSA and CS (-0.53). This means that the evaluation of DESvisual by CS majors is opposite to that of the CNSA majors. In fact, evaluation by CS majors has a negative correlation with other majors, although it

is weak and may be considered as insignificant. On the other hand, CpE major responses showed noticeable positive correlations with CNSA and Math, and the correlation between CNSA major responses and those of Math is insignificant or uncorrelated. In summary, this study shows CS majors reacted to DESvisual in a very unique way, responses from CpE majors have noticeable positive correlation with those CNSA and Math majors, and responses from CNSA majors and Math are uncorrelated.

Table 3: Correlation Matrix

	CNSA	CpE	CS	Math
CNSA	1	0.19	-0.53	0.03
CpE	0.19	1	-0.17	0.16
CS	-0.53	-0.17	1	-0.04
Math	0.03	0.16	-0.04	1

The set of 12 write-in questions is designed to allow students making detailed descriptions and suggestions which will be used for future development. We focus on the following issues: (1) whether a single round of the Feistel cipher is good enough, (2) whether a restriction to 8-bit and 16-bit for demo and practice is sufficient, (3) the design of the Feistel cipher, S-box, and decryption, (4) evaluation of the demo and practice modes, (5) frequency of using DESvisual for self-study, and (6) software installation problems.

Student comments show that one round of the Feistel cipher and 8-bit and 16-bit for demo and practice is good enough. Only very few suggested to have more than one Feistel rounds, and 12-bit and even 64-bit input. There was no negative comments on the design of the Feistel, S-box, and decryption. Typical comments are "The S-box design seems like a very novel way to prevent patterns from emerging between the plaintext and ciphertext", "This design helped me understand how the S-boxes were used", "[The Feistel cipher] design was adequate to show flow of data and lines of data dependence" and "[The Feistel cipher] is straight-forward (easy to follow), works fine". There were, of course, some issues raised, mostly from non-CS related majors. Major issues were (1) should support 12-bit input and

a full DES cipher, (2) should use symbols and notations exactly the same as the simplified DES algorithm in the textbook [8], and (3) the designs are overly complex.

Student comments were also positive about the bit trace feature, practice mode, and the usefulness of the demo mode for classroom presentation. For the bit trace feature, students indicated "I loved that I could click a bit and trace it to its transformed place", "It makes me every sense for what is going on", and "I enjoy this feature the most". Not all students like bit trace, though. A few students suggested that bit trace is somewhat useful or too complex. The practice mode is also liked by most students with comments like "The added flexibility of the practice mode gave a little more 'hands on' learning", "practice mode functions as I expect", and "practice mode was super helpful as it guided me step by step through the process". Reactions to the demo mode is a bit split. Most students mentioned that the demo mode is useful with typical comments like "I feel that the demo version allowed me to follow the algorithm somewhat better" and "alternative means of presentation was helpful in solidifying understanding". On the other hand, a significant number of students indicated that "the visuals were incredibly helpful but less classroom participation" and "blackboard and visual are equally effective". This verifies a common sense that visualization cannot be used alone and must be accompanied with an effective classroom presentation.

Because the students only had a week to play with DESvisual before taking this survey, the frequency of using this tool is not very high. Most of them used the tool a few times, and a few of them played with the tool "quite often". In general, they used DESvisual when they were solving problems, checking for some details, forgot the inner working of the algorithm, and, of course, used it for practice and further understanding. Since the DES algorithm is not very complex, we are not surprised by the fact that a few students only used it once or twice, or did not use it at all.

As for installation, students did not report any issues; but, a few of them liked the MacOS version better than the other two, and suggested to add more help screens.

In summary, with the statistics and student comments presented above, we believe DESvi-

sual has fulfilled its purpose, helped students learn and the instructor teach the DES component. With the comments and suggestions, we should be able to improve DESvisual significantly in the near future.

5 Conclusion

The above presented a visualization tool DESvisual for teaching and learning the DES algorithm. With DESvisual instructors are able to present all the details and inner working of the DES algorithm. It also helps students see the "flow" of the algorithm, trace a selected bit to its transformed place, and learn the computation steps using the practice mode.

Our survey results showed that DESvisual was effective in classroom presentation and for student self-study. Students in this class were in five disciplines with computer science majors being the largest. We also investigated the impact of DESvisual on each group. Our findings indicated that the computer science group reacted to DESvisual in the most positive way, and had a minor negative correlation with the other groups. Whether this is a disciplinary difference is not clear at this point and further investigation is required. However, the positive impact of DESvisual on computer science students learning is undeniable.

DESvisual is a part of larger development project of cryptography visualization tools. We are currently working on the visualization component for elliptic curves and finite field based cryptography. In the near future, DESvisual will be extended in a number of ways. The most needed extensions are (1) the support of variable number of bits input rather than fixed to 8-bit and 16-bit, (2) a better and broader bit trace mechanism and GUI design, (3) a better S-box design, and (4) a library to support DES based programming assignments. Moreover, we plan to design a class library with which student programs may turn on or off the visualization feature without extra instrumentation so that the visual aid may also be used as a debugging tool. This technique was used in ThreadMentor successfully [2, 7].

References

[1] David Bishop. *Introduction to Cryptography with Java Applets*. Jones and Bartlett, 2003.

[2] Steve Carr, Jean Mayo, and Ching-Kuang Shene. ThreadMentor: A pedagogical tool for multithreaded programming. *ACM Journal on Educational Resourses in Computing*, 3(1), March 2003.

[3] Oi-Shong Chok and Susantha Herath. Computer Security Learning Laboratory: Implementation of DES and AES Algorithms using Spreadsheets. In *MICS 2004 Proceedings*, April 2004.

[4] Alasdair McAndrew. Teaching Cryptography with Open-Source Software. In *ACM 39th SIGCSE Technical Symposium*, pages 325–329, 2008.

[5] Michael Rosing. *Implementing Elliptic Curve Cryptography*. Manning, 1999.

[6] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Wiley, second edition, 1995.

[7] Ching-Kuang Shene. Multithreaded Programming with ThreadMentor: A Tutorial, 2001. `http://www.cs.mtu.edu/~shene/NSF-3/e-Book/index.html`.

[8] Wade Trappe and Lawrence C. Washington. *Introduction to Cryptography with Coding Theory*. Prentice Hall, second edition, 2005.

[9] Michael Welschenbach. *Cryptography in C and C++*. Friend of ED, third edition, 2005.