3D Digital Legos for Teaching Security Protocols

Li Yu, *Student Member*, *IEEE*, Lane Harrison, *Student Member*, *IEEE*, Aidong Lu, *Member*, *IEEE*, Zhiwei Li, *Student Member*, *IEEE*, and Weichao Wang, *Member*, *IEEE*

Abstract—We have designed and developed a 3D digital Lego system as an education tool for teaching security protocols effectively in Information Assurance courses (Lego is a trademark of the LEGO Group. Here, we use it only to represent the pieces of a construction set.). Our approach applies the pedagogical methods learned from toy construction sets by treating security primitives as Lego pieces and protocols as construction results. Simulating the Lego toys, the digital Legos use matching shapes to help students understand the relationships among security primitives and protocols. Specifically, we present a flexible Lego generation method that can use various intuitive shapes to represent abstract and complex security protocols. Our design allows easy generation of new Lego sets and creation of different course materials. The integrated system also provides 3D interaction methods that simulate the real Lego building experience. For selected security courses, we have designed sample demonstrations and experiments for a set of important protocols. The initial evaluation results show encouraging feedback from students on using digital Legos in introductory security courses.

Index Terms—Security protocol, digital Lego, construction set, visualization for education.

1 Introduction

Information assurance education for both college students and the general public has been well recognized by many universities as an important topic since the early nineties. For example, Pothamsetty has investigated 25 security courses offered by multiple universities that are designated as NSA Centers of Academic Excellence, and found that most of them adopted a curriculum structure of introductory-advanced information assurance courses [1]. Our experiences in teaching introductory and advanced security courses have led us to realize that there exists a gap between the teaching of security primitives and protocols, which may severely impact the learning outcomes of information assurance education. It has been shown that a group of secure primitives may finally compose vulnerable protocols if they are inappropriately organized. Therefore, special efforts must be made in the course plan to cultivate the capability of students to select suitable primitives and organize them appropriately. We believe that an interactive education environment for demonstration and exercises can help bridge this teaching gap.

The objective of this project is to develop an innovative digital construction set by integrating the achievements in security education and visualization. We also design instructional demonstrations and hands-on experiments, using the set to assist students in bridging security primitives and protocols. Our approach applies the pedagogical methods that have been learned from the success of children and adult education, using electronic blocks or construction sets [2], [3]. Specifically, we treat security primitives as Lego pieces and protocols as construction results. Our work can serve two tightly integrated purposes: automatic demonstrations of protocol decomposition to help students understand the relationships among primitives and protocols, and hands-on experiments to cultivate their capabilities to manipulate primitives and design protocols that satisfy different security requirements. The latter is one of the ultimate objectives of information assurance education.

The main contribution of this research is a 3D digital Lego approach that visualizes security protocols effectively and automatically to teach information assurance courses. Compared to traditional methods, this approach attempts to better reveal the relationships among security primitives and protocols, thereby improving security education outcomes. Our design of 3D digital Legos allows other instructors to develop, share, and modify the sample Lego sets so that they can generate their own demonstration and experiment materials easily. Based on this approach, we have developed a prototype system with several important interaction functions that can be used as a user-friendly demonstration and experiment environment. We have also performed initial evaluations to assess this Lego-based approach on teaching introductory security courses and received positive feedback.

The paper is organized as follows: Section 2 discusses related work on construction sets in education and graphical approaches for information assurance courses. In Section 3, we present our efforts to explore suitable representations of security protocols using real Lego toys, which help us design the 3D digital Legos. Section 4

- *L. Yu, L. Harrison, and A. Lu are with the Department of Computer Science, College of Computing and Informatics, University of North Carolina at Charlotte, 9201 University City Blvd, Charlotte, NC 28223. E-mail: {lyu8, ltharri1, aidong.lu}@uncc.edu.*
- *Z. Li and W. Wang are with the Department of Software and Information Systems, University of North Carolina at Charlotte, 9201 University City Blvd, Charlotte, NC 28223. E-mail: {zli19, WeichaoWang}@uncc.edu.*

Manuscript received 16 Feb. 2010; revised 4 June 2010; accepted 11 July 2010; published online 27 July 2010.
For information on obtaining reprints of this article, please send e-mail to: lt@computer.org, and reference IEEECS Log Number TLT-2010-02-0016.
Digital Object Identifier no. 10.1109/TLT.2010.19.

TABLE 1
Notations of the Symbols

representation	meaning
A	entity A
N_B	random number generated by B
K_u_B	the public key of entity B
K_v_B	the private key of entity B
K_s_AB	the symmetric key shared between A and B
$A \rightarrow B$	A sends a message to B
x, y	concatenation of items x and y
$\{msg\}key$	a message encrypted with the key

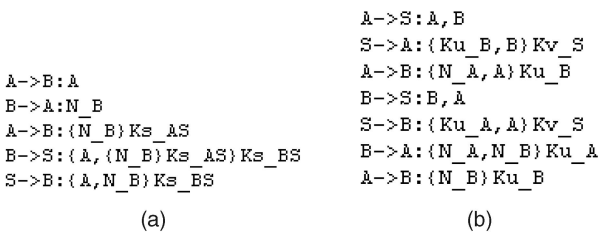


Fig. 1. Two example security protocols, (a) Woo Lam Protocol and (b) PKP Protocol, shown in plain text.

describes our approach to 3D digital Lego generation for visualizing security protocols. Section 5 presents the integrated system as a user-friendly demonstration and experiment environment. We describe our evaluation processes and results in Section 6 and provide a discussion on our approach in Section 7. Finally, Section 8 discusses future extensions and concludes the paper.

2 Related Work

2.1 Construction Sets

Construction sets have a venerable place in the history of education. Records show that as early as in 1,800 appeared a building set for castles and walled towns [4]. In America, building blocks have been recommended to parents since 1826 [5].

Recently, the educational role of construction sets has been enhanced by the integration of computational media. For example, building blocks with sensors and fiber optic output were used to construct a speech-enabled alphabet set [6] or 3D structures for communicating to a computer [7]. Particularly, construction sets have been widely used in undergraduate robotics education. For example, Lego bricks [8] were used as the controllers for large Lego sets. The sets provided a wide space for students to make hypotheses about how things work and validate their assumptions [9]. Similar digital manipulations have been used in artificial intelligence, programming, and general engineering courses [10], [11], [2]. Inspired by the success in robotics education, digital construction sets have been applied to the design of space habitat and vehicle [12] and computer systems [13]. For example, the functional decomposition approach [13] has been applied to many systems, including analog electronics, digital design, VLSI, and software.

In this paper, we present an approach that adopts the concept of Legos to help students understand the relationships among security protocols and the involving primitives. Different from previous methods, our approach can automatically generate specialized digital Legos for various security protocols.

2.2 Achievements in Security Education

This project is inspired by the fact that various security protocols are constructed by a limited number of primitives. For example, Millen et al. [14] have summarized ten reduction rules to decompose security protocols into simple units, and Cremers [15] has investigated how to decompose a complicated protocol into subprotocols. Therefore, we believe that a suitable design of digital Legos can be used to assist us in teaching security protocols. Previously, we have developed a 2D Lego system for security courses [16], in which, special 2D Lego pieces are designed to visualize the operations such as encryption. Our experiences show that the 2D shapes may cause some difficulty in understanding the security protocols, since the message contents are represented as embedded boundaries. Therefore, in this paper, we present a more intuitive approach that can simulate the real 3D Legos.

Several other graphical approaches have been proposed for security education. For example, Burger and Rothermel presented a general purpose simulation architecture for teaching security protocols [17]. Saul and Hutchison developed a graphic environment for analyzing security protocols [18]. Schweitzer [19] designed an interactive visualization tool for demonstrating protocols, visually in a user-controlled stepwise manner. Elmqvist also developed an animation function to display protocols in a step-by-step fashion [20]. In contrast to our approach, these methods are mainly designed to use graphics or interactions to emphasize the sequential events associated with a protocol. Compared to previous methods, our approach can illustrate the messages of a protocol in visual forms and demonstrate the relationships among primitives and protocols.

3 Construction with Real Legos

3.1 Notation

We first introduce the notations that are used to describe a security protocol in the remainder of the paper. A security protocol usually consists of the interactions among multiple entities. We adopt the Dolev-Yao model [21] to represent the deduction capabilities of the legitimate entities and attackers. Table 1 lists the notations of the security protocols.

To build a generic approach that can represent a wide range of security protocols and attacks, we have adopted a flexible two-tier construction method [16].

We use the subindex of an item to label its owner so that the end users of our system can directly edit the protocol files. For example, N_B represents a random number generated by entity B. If the subindex contains two entity names, it is shared between them. For example, K_s_AB represents a secret key shared between entity A and entity B. As illustrated in Fig. 1, this approach represents messages of security protocols with plain text and they can be easily understood by the end users.

3.2 Protocol Construction with Real Legos

Before designing 3D digital Legos, we have explored several ways to use real Lego blocks to construct security

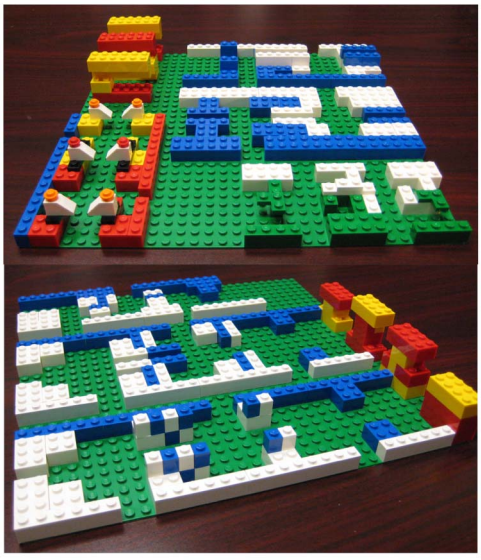


Fig. 2. Example results of security protocols built with real Legos. Five designs are shown on the top and four are shown on the bottom.

protocols. This experiment helps us learn how the concepts of Legos can be used to represent primitives and protocols, and assists us in designing effective 3D digital Legos for security education. The results also confirm our hypothesis that Legos can be used as an appropriate metaphor in an education tool to expose the relationships among security primitives and protocols. Below, we describe our selections of protocol representation and the designs of protocol construction.

For constructing various protocols using real Legos, we have selected a Lego product that satisfies two requirements. First, we look for products that contain small Lego pieces so that the final construction results are in an appropriate size for demonstration and storage. Second, we need a large number of Lego blocks with similar shapes, since primitives usually appear multiple times in a protocol. Under these two requirements, we have selected the "Lego System Ultimate Building Set" made by LEGO as our tool.

We have explored several ways to construct security protocols with real Legos. We use colors to differentiate entities. For example, in Fig. 2, red and yellow, blue and white, or green and white are used to visualize entities A and B, respectively. We choose one or several Lego blocks to represent the primitive types. To utilize the available Lego pieces efficiently, we select combinations of Lego shapes for different primitive types carefully through the following procedure. First, we summarize the frequencies of primitives in several security protocols that are taught in our introductory level security course. Then, the number of each Lego shape is counted. By matching the numbers of available Lego blocks to the frequencies of primitives, we ensure that our design can utilize the available Lego blocks efficiently.

Based on the designs of primitives, we have explored several methods to construct protocols. Fig. 2 shows five designs for the Needhand-Schroeder-Lowe protocol on the left and four designs for the Andrew Secure RPC protocol on the right. Our main choices are between the vertical and flat designs for the message contents. For example, the top left red-yellow design in Fig. 2a is a vertical version for providing a strong transition impression, and the blue-white designs in Figs. 2a and 2b are flat versions for demonstrating message contents. It is interesting to note that multiple ways can be used to construct a protocol even with a simple Lego set. Also, this experiment helped the authors to remember several security protocols easily.

4 Automatic Construction of 3D Digital Legos

We design a method to construct specialized 3D digital Legos, automatically, for teaching security protocols. This method allows more flexible generation of instructional demonstrations and hands-on experiments than real Legos. Compared to the traditional text-based methods (examples shown in Fig. 1), our Lego-based approach provides more effective course materials to direct the students' focus and attract their interests.

In this section, we present a generic method to construct 3D digital Lego sets for teaching various security protocols. Our method is developed based on the two-tier protocol representation that enables our approach to visualize different security protocols and attacks. The entire generation process is automated to allow easy creation and sharing of course materials.

4.1 Basic Lego Design

To better expose the relationships among primitives and protocols, we use different shapes to represent the primitive types and different colors to represent the entities. For each Lego block, only one surface is chosen to carry the information of message contents and is used to determine whether or not two blocks can fit together. In this way, a protocol can be visualized as multiple sending and receiving blocks.

Specifically, our digital Legos are constructed with the following procedure. First, we generate a set of geometry meshes to represent the primitive pieces based on 2D designs. Second, multiple blocks of digital Legos are composed in an appropriate order to visualize a security protocol.

Since we want to construct the digital Lego blocks automatically for a given protocol, we use two portions with fixed shapes and two portions with adjustable shapes to compose one Lego block. As shown in Fig. 3, the top, bottom, and body define the general shape of a Lego block and the content surface is generated according to the message content. The shapes of the top and bottom portions match each other to ensure the vertical connection between any two blocks. They always point downward, since we assume that the protocols are executed from top to bottom. The content surface carries the most important information, so we use a later section to discuss its generation in detail. The length of a block is also automatically adjusted according to the content of a message.

Our main purpose for separating the sending and receiving blocks is to provide flexibility to the demonstration and experiment tasks. Although a message is shared between a sender and a receiver, their interpretation of the same message may be different, especially when attackers are involved. This also allows us to show different detail levels of the same message in demonstration and experiment tasks.

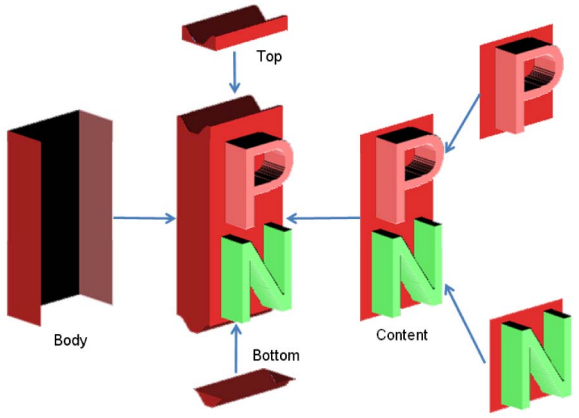


Fig. 3. Our Lego block consists of two portions with fixed shapes (top and bottom) and two portions with adjustable shapes (body and content). The adjustable portions are automatically generated according to the message content and sending/receiving type.

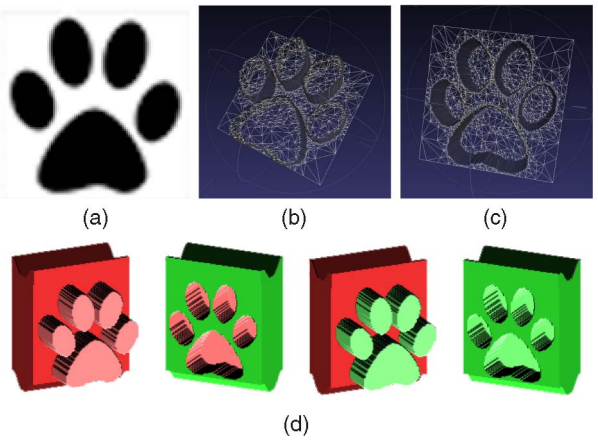


Fig. 4. The generation process of primitive pieces. (a) We design gray-scale images to represent primitive shapes. (b) The input image is automatically converted to a 3D mesh to represent the sending operation. (c) We reverse the point heights in (b) to represent a receiving operation. (d) Sample results of using this primitive piece during Lego visualization.

Once a protocol is selected, this Lego design allows us to generate all the Lego blocks automatically. For better discretion of the Lego body and the message contents, we use a similar but deeper color for the Lego body. We believe that this design matches the spirit of Legos closely, which is to capture the interests of students and attract their focus to important security concepts represented by functional Lego shapes.

To generate 3D Legos, we use polygon meshes because of their flexibility. As shown in Fig. 3, the top, body, and bottom portions are composed of simple 3D meshes. We use the following two sections to describe our procedures to generate the content surfaces.

4.2 Design and Generation of Primitive Representations

Since the message surfaces are rendered in 3D, we also prepare our primitive pieces in 3D, so that they can be used to compose 3D Lego blocks quickly during the rendering process. The following describes our method that allows users to design the shapes of primitives by transforming 2D images to 3D meshes.

Our method allows users to design their primitives using gray-scale images, as shown in Fig. 4a. Instructors can use any image editing software to input their design easily, and the rest procedure is automatically handled by our method. Specifically, we map a $n \times n$ (100 is used for all the examples in this paper) grid on the input image, and preserve all the line connections in the grid. The pixel colors (darkness values) in the image are used to adjust the corresponding point heights in the grid. This procedure generates an initial 3D mesh that matches the appearance of the input image. We also use the point heights to separate the raised portions from the background and assign them to different colors during visualization. To improve the efficiency of the rendering process, we simplify these meshes with the MeshLab software [22] to generate the final primitive pieces. Fig. 4b shows the generated mesh based on Fig. 4a and the protruding surface is used to represent a sending operation. We reverse the mesh in Fig. 4b to generate the receiving piece with a dented surface. This design ensures that two content surfaces can be put face-to-face if, and only if, their shapes match. The content surface is then combined with the other portions of a Lego block to generate the final results in Fig. 4d.

We have designed several Lego sets to cover all the primitives in our selected uniform representation of protocols, as shown in Fig. 5. These results demonstrate that our approach can generate various Lego sets flexibly. This method also allows other users to share these designs and create their own shapes easily. We believe that the ability to switch primitive designs can help users to choose their desired styles and make the learning process more attractive to students.

4.3 Generation of Content Surfaces

With the primitive pieces created above, we can automatically generate the content surfaces of Lego blocks for a given message. To compose a connected 3D mesh as the content surface, we use the following procedure, which first arranges a message content on a 2D table and then stitches corresponding primitive pieces together.

A message often consists of a list of primitives connected by manipulation operators such as encryption and concatenation. We can view concatenation as the connection of two or more primitives at the same level, and encryption as the coverage of primitives at a deeper level. A 2D table can be generated for any given message. For example, Fig. 6a shows the filled 2D table for message "{B,{N_A,N_A}Ks_BS, A}Ks_AS" in the Yahalom protocol. Starting from a corner of the 2D table, we fill it with the message content by increasing the row when seeing concatenation or increasing the column when seeing encryption. We also record whether or not a location on the table has content or not by assigning a 0/1 flag to it. In this way, we can use such a 2D table to represent any message.

During the real-time rendering process, we draw Lego blocks according to their content tables. For locations without any content (with flag 0), we draw one big polygon to cover the space. For locations with contents (with flag 1), we draw the corresponding primitive pieces in the preassigned colors

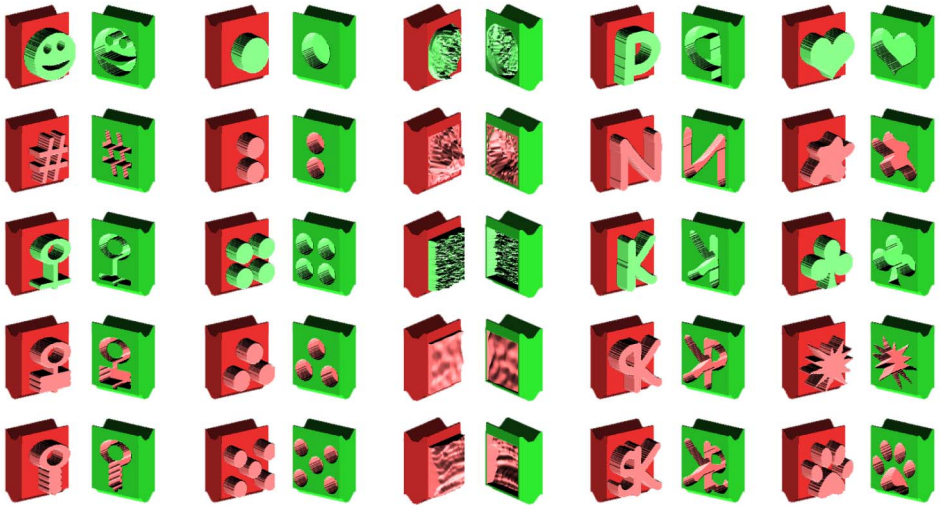


Fig. 5. Examples of our primitive designs. From top to bottom are principal/entity, random number, public key, private key, and symmetric key. The pieces with red bodies are sending blocks and the green bodies are receiving blocks.

of their entities. For symmetric keys, we take the colors of both entities and use each to draw half of the mesh. Fig. 6 shows an example of our rendering method.

Here we use the Woo and Lam Pi 3 protocol as an example to illustrate our rendering process. Fig. 7 shows the traditional text version of the protocol on the left, and the digital Lego version on the right. We use colors to represent the communicating entities in the protocol: red for Alice, green for Bob, and blue for the server. In this example, we choose the last style shown in Fig. 5, with the heart shape representing an entity, the star representing a random number, the club representing a public key, the claw representing a symmetric key, and the echinus shape representing a private key. Each row of the rendering represents the transition of one message. For example, the first row is Alice sending her identity A to Bob. The convex and concave shapes are used to indicate the sending/receiving operations.

5 Integrated Lego System

With our digital Lego sets, we have developed two types of course materials: protocol demonstrations and hands-on experiments. The demonstrations are designed to better illustrate important protocol concepts during lectures. As a complementary component, our hands-on experiments are developed to train students to apply security knowledge flexibly during protocol design. We have integrated both components into one prototype Lego system, so that students can study examples and take exercises with the same tool.

5.1 System Design

We develop our system with a multipanel interface design. As shown in Fig. 8, our system is composed of a main rendering window on the left and two interaction windows on the right. The main window contains four panels: primitive panel for displaying the current primitive design (left top), protocol panel for node knowledge, protocol contents or exercises (left middle), attack panel for attack strands and the knowledge of a node selected by users (left bottom), and rendering panel for visualizing and interacting with 3D Legos (right). The right top window is designed for users to adjust the rendering and interaction settings. The right bottom window is for controlling the exercise process. This multipanel interface allows us to integrate multiple demonstration and experiment functions into our Lego system.

1 (Ks_AS)	1 (B)	0
0	1 (Ks_BS)	1 (N_A)
0	0	1 (N_A)
0	1 (A)	0

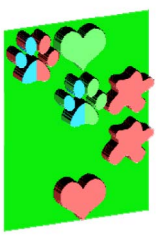


Fig. 6. The generation process of a content surface. A message "{B,{N_A,N_A}Ks_BS,A}Ks_AS" is first converted to a 2D table (left), then rendered with corresponding primitive pieces automatically (right).

```
A->B:A
B->A:N_B
A->B:{N_B}Ks_AS
B->S:{A,{N_B}Ks_AS}Ks_BS
S->B:{A,N_B}Ks_BS
```

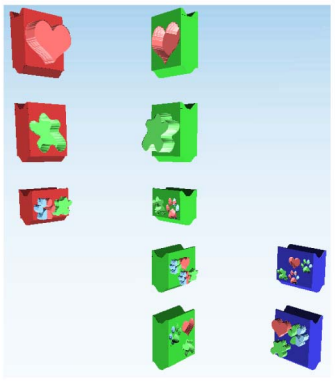


Fig. 7. Woo and Lam Pi 3 protocol shown in the text-based approach (left) and the Lego-based approach (right).

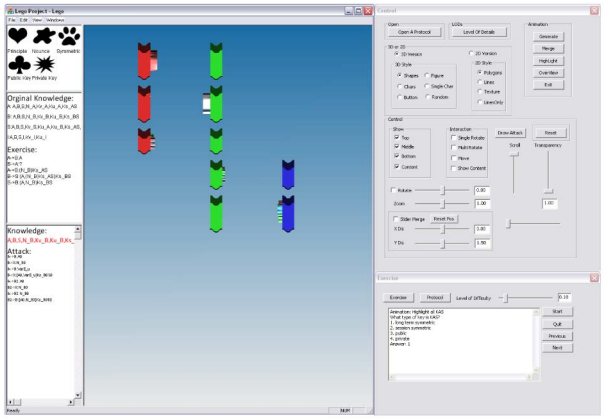


Fig. 8. Our system interface includes both 3D Lego-based and text-based interaction panels.

5.2 Interaction

Since we visualize security protocols with 3D digital Legos, it is important to provide suitable interactive methods that allow users to browse Lego contents freely and assist them in constructing protocols. As examples show in Figs. 9 and 10, our system is capable of the following specialized interaction functions:

- Rotating: Viewers can rotate individual or a group of Lego blocks.
- Moving: Viewers can also move individual or a group of Lego blocks around the screen space.
- Displaying messages: The message contained in a selected Lego block is displayed in a floating window.
- Facing-to-viewer: We design a special facing-to-viewer rendering function, which turns the message content surfaces of selected Lego blocks to viewers while preserving the central positions of these Lego blocks, as shown in Fig. 9.

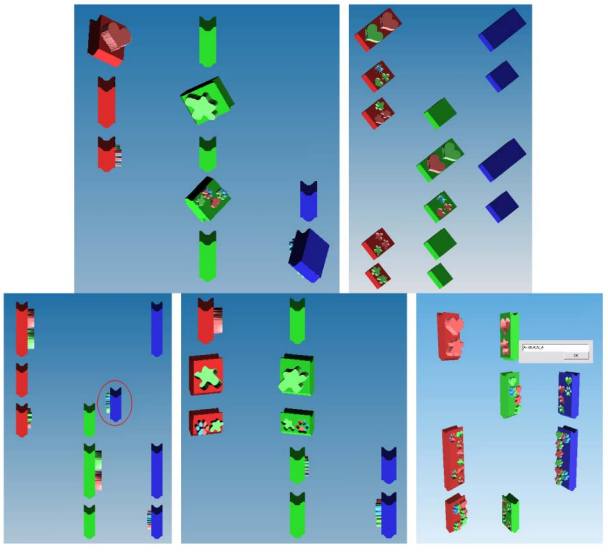


Fig. 9. Interaction examples of the single rotation, multiple rotation, moving, facing-to-viewer, and displaying message functions.

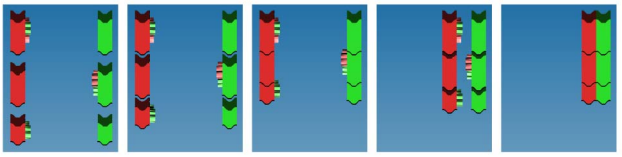


Fig. 10. An example of the merging interaction. The Lego blocks in the same column are sent by the same entity and they are merged first. Then all the columns are merged.

- Merging: Viewers can adjust the distances between adjacent Lego blocks and finally merge all of them, just like playing with real Legos. An example of this merging function is shown in Fig. 10.
- Labeling: We also allow viewers to select important primitives or protocol portions and adjust their rendering parameters to emphasize the important contents.

5.3 Experiments

We also design an experiment function for our digital Lego system so that it can be used for practice and homework. An experiment panel is provided with the supporting functions, such as start, next, and complete. Suitable rendering settings are also adopted during the visualization process. For example, Fig. 11a shows a Lego block in our filling experiment. This experiment is designed to emphasize important portions of protocols and strengthen related concepts by asking students to complete a predesigned protocol. For each of our sample protocols used in the class, we randomly remove a portion of Lego blocks or messages that are related to the lecture contents. A difficulty level is used to control the amount of information that is hidden. As shown in Fig. 11, our Lego construction method can automatically visualize the protocol. We can also change the settings of 3D Legos to direct the attention of users to specific portions of a protocol, such as using the facing-to-viewer motion in Fig. 9.

5.4 Results

We have tested the Lego approach with all the security protocols being taught in our undergraduate course "Introduction to Information Security and Privacy." The selected protocols include the examples that are widely used in security courses, such as Woo Lam protocol, Neumann Stubblebine, Needham Schroeder Public Key, Needham Schroeder Lowe Public Key, and Otway Rees. We have also selected several protocols from real-life applications, such as BAN modified version of CCITT X.509 (3), Kerberos V5, and KSL (Nonce based improvement of Kerberos V5). Many of these protocols have been collected by the SPORE project [23]. For all these protocols, our

Fig. 11. Experiment examples. Selected primitives can be automatically replaced with a "?" mark.

Fig. 12. Example results demonstrate that our approach can visualize various security protocols.

approach can automatically generate the Lego-based protocol visualizations. Fig. 12 shows the Lego representations of the following messages:

- "A->S:{N_B}Ku_B"
- "A->S:A"
- "A->S:{Ks_AB,N_B,A}Ks_BS"
- "S->A:{N_A,B,Ks_AB,{Ks_AB,A}Ks_BS}Ks_AS"
- "A->S:B,N_B,{A,N_A}Ks_BS"
- "A->S:A,N_A"
- "S->A:{B,Ks_AB,N_A,N_B}Ks_AS,{A,Ks_AB}Ks_BS"

Due to the space limit in Fig. 12, we illustrate messages with different lengths and complexity, instead of the entire protocols. Since a protocol is composed of individual messages, these examples demonstrate that our Lego construction approach can handle quite a variety of security protocols.

Fig. 13 shows one message represented in several designs of security primitives. Other instructors can either adopt our samples directly or design their own appearances of primitives. Our approach to constructing digital Legos allows an easy switch of primitive designs during runtime.

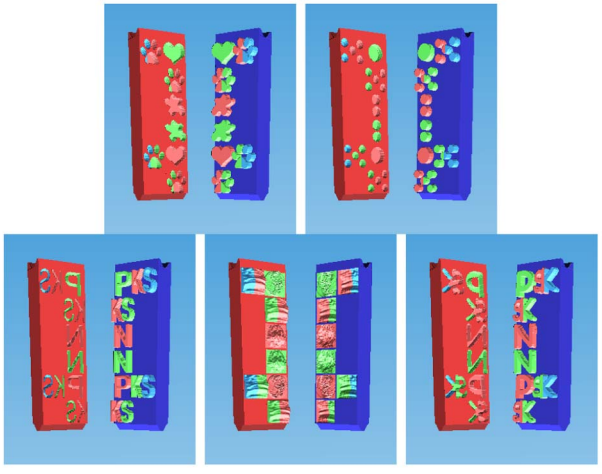


Fig. 13. The message, $S \rightarrow A : \{B, Ks\_AB, N\_A, N\_B\}Ks\_AS, \{A, Ks\_AB\}Ks\_BS$, visualizes in different primitive styles. Our Lego construction approach can switch among different primitive styles in real time.

We have also tested the usages of our interaction functions. When studying a protocol, we often use the merging function to visualize the entire protocol. Then, we use the facing-to-viewer function to browse the contents of individual messages. For protocols with many rounds of interactions, we can use the moving function to scroll down the screen to view the entire protocol. For a particular Lego block, viewers can use the rotating function to observe details or use the labeling function to view the text representation of the message. We believe that these functions are essential to help instructors or students to experience realistic interaction with 3D Lego-represented security protocols.

6 Evaluation

We have designed and performed user studies to evaluate the effectiveness of our Lego-based approach on teaching security protocols. The main evaluation goal was to compare our Lego-based approach with the traditional text-based approach from different educational aspects. The results of these studies have provided important information to us on the advantages of the new Lego-based approach, as well as useful clues to improve this visual-based scheme.

Our evaluation plan consists of the following two portions: an informal survey for gathering feedback on the general Lego-based approach, and a formal user study for assessing the specific performance of the Lego-based and text-based approaches. The following first describes the informal survey, which shows significant interests in using Legos in class from students. Later, we present two experiments in our user study and discuss their results.

6.1 Survey

Due to the limit of available time in our class, we designed a brief survey to assess the general interest of students for Lego-based approaches. Our hypothesis is that an interactive tool based on a popular toy concept would pique the interest of students in computing majors more so than traditional text-based methods. We believe the positive results indicate that visual approaches can better encourage students to study challenging and abstract security theories. The following lists the subjects, procedure, results and discussions of our survey.

6.1.1 Subjects

Our subjects include 23 student volunteers from the "Introduction to Information Security and Privacy" class

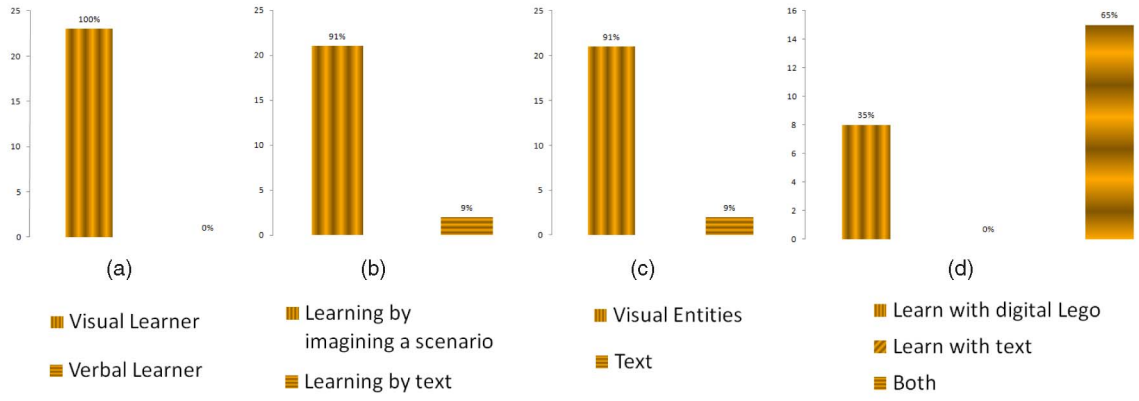


Fig. 14. Multiple choice questions in the survey and their results shown in the bar graph. The questions are: (a) "Do you consider yourself as a visual learner or a verbal learner?" (b) "If I were learning about a protocol, I would prefer to walk through each step by imagining a scenario where two or more entities execute the steps or use the text-based methods used in class so far?" (c) "When I think of a security protocol, I get something most like visual of entities or text of a protocol?" (d) "I feel I can learn best by using digital Legos, text or both?" The colors represent the choices of students.

at UNC Charlotte. The majority of our subjects are juniors with computing backgrounds. Since these students have learned security protocols using the text-based approach throughout the semester, they are all equipped with basic knowledge of security protocols and are familiar with the text-based approach.

6.1.2 Procedure

Before the survey, a 15 minute introduction of our Lego-based approach is given to the subjects. Since these students are familiar with the text-based approach, we concentrate on explaining how the designs of digital Legos can be used to teach the primitive and protocol relationships in general security protocols. We also demonstrate the digital Lego system and its interaction methods. After the introduction, we answer questions raised by the students for about 10 minutes.

During the survey, each student is given a copy of the survey questions and instructed to take as much time as they need to finish. The survey is in the form of multiple choice, Likert-scale, and free response questions. The questions are used to assess the interest of students on general Lego related issues. Fig. 14 shows the four multiple choice questions, and Table 2 shows the four Likert-scale questions (the six scales are strongly agree, agree, slightly agree, slightly disagree, disagree, and strongly disagree).

6.1.3 Results and Discussions

The results of this survey indicate a strong motivation of students to combine text and visual based approaches to learn security protocols. Considering that the students participating in this survey have only been introduced to the Lego-based approach shortly, we think that they may have questions and concerns on the details of digital Legos. Even so, a majority of the students still choose the Lego-based approach in addition to the traditional text-based approach.

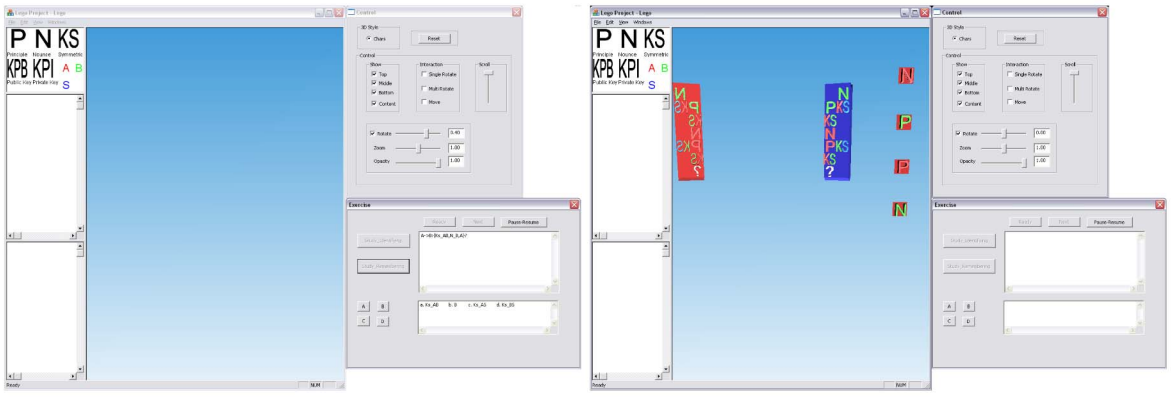
As shown in Fig. 14, all the students consider themselves to be visual learners, which shows a unanimous interest in improving the traditional text-based approach. About 91 percent students prefer to learn security protocols as visual entities and imagine protocol scenarios in visual forms. This number indicates a wide acceptance of visual-based education tools. Also, in the last question "I feel I can learn best by digital Lego, text, or both?," none of the students chose the text-based approach and about 65 percent chose a combined Lego and text-based approach. Since our system is able to show the plain text of protocols as well as digital Legos, our system design matches the interests of students.

Table 2 shows our Likert-scale questions and their results. If we use the scores 0 to 5 to represent the choices "Strongly Disagree" to "Strongly Agree," the averages are 4.3, 3.35, 3.45, and 2.75 for questions (a)-(d), respectively. For the first question, "I feel I can learn protocols by a visual approach," the average score 4.3 shows a strong confidence in visual-based approaches. For the last question, "I feel I can learn security protocols by a text-based approach," the average score 2.75 is just a little bit higher than neutral. Since these students have been taught with the text-based

TABLE 2
The Results of the Likert-Scale Questions in the Survey

Questions	Strongly Agree	Agree	Slightly Agree	Slightly Disagree	Disagree	Strongly Disagree
(a)	9	12	2	0	0	0
(b)	3	8	9	0	3	0
(c)	10	4	2	3	2	2
(d)	0	5	11	3	4	0

The questions are: (a) "I feel I can learn security protocols by a visual approach." (b) "I feel I can learn security protocols by a digital Lego system." (c) "In the past, I played with Legos a lot." (d) "I feel I can learn security protocols by the text-based approach." The numbers represent the choices of students.



(a) (b)

Fig. 15. The interface of our experiment environment. (a) shows a question "$A \rightarrow B : \{Ks\_AB, N\_B, A\}?$" using the text-based approach with the questions and choices displayed in the right bottom panel. (b) shows the same question using the Lego-based approach with the questions and choices shown in the middle 3D Lego panel.

approach, we think that this score indicates some obvious obstacles they have experienced during the semester. These two numbers match the results in Fig. 14 as well. For subquestions (b) and (c), we can see high percentages of students who have played with Lego in the past and who feel that they can learn security protocols by a digital Lego system. We find that four out of the seven students, who have not played with Legos, are also interested in the visual aspect of this approach. This result indicates that the combination of digital Legos and text in our system may best serve for the purpose.

6.2 User Studies

The survey results support our contention that visual-based approaches should be used to improve the teaching of security protocols. We have further designed and performed two user studies to assess our Lego-based approach. Specifically, we concentrate on two essential aspects of learning: identification and memorization of security primitives and protocols.

We modify our digital Lego system to generate an experiment environment for our user studies. The following describes the three major changes:

- Adding automatic experiment functions, including randomizing question sequences, recording individual operation time durations and user answers, allowing pause and resume during the experiment, and saving subject files;
- Enabling and disabling experiment buttons for different experiment phases. During the observation phase, only one "question" button is active for viewing questions; during the response phase, only the multiple choice buttons are active; and the "next" button becomes active only after an answer has been selected. This function guides the subjects to finish the experiment without distraction.
- Adjusting the user control panels by hiding all unnecessary interaction buttons.

Fig. 15 shows the interface of our experiment environment using the Lego-based approach and the text-based approach, respectively. The control buttons used during the studies are the same for both methods, so that they do not affect the study results.

Before the experiments, we hold a practice session to familiarize the subjects with our experiment environment and procedure. The procedure of the practice session is the same as our user studies, except that the practice session only contains one sample question and explanation for each experiment. This practice session is designed to reduce the confusion of subjects during experiments and ensure the accuracy of our captured time durations.

6.2.1 *Experiment 1: Protocol Primitive Identification*

Since the survey results have shown that the Lego-based approach can attract the attention of students, we are interested in finding out how this approach can assist the teaching of security protocols. Our first hypothesis is that the Lego-based approach could help students identify important primitives in a protocol more easily than the text-based approach. We design this experiment to evaluate the aspect of identification through measuring the factors of accuracy and time duration during identification tasks.

Apparatus: A Windows machine with an ordinary USB mouse.

Subjects: Seventeen students (5 female and 12 male) volunteered from the "Introduction to Information Security and Privacy" class. They have all taken the survey before this experiment.

Materials: Since this experiment requires subjects to study an entire protocol carefully, we have selected four short protocols: one contains seven messages and the other three contain five messages each. Also, all of the messages in these protocols consist of a small number of primitives. Fig. 16 shows these four protocols, corresponding multiple choice questions, and their answers.

Procedure: To avoid the factor of question orders influencing the user study results, we adopt the following

Question 1:
A->S:A,B
S->A:{Ku_B,B}Ks_AS
A->B:{N_A,A}Ku_B
B->S:B,A
S->B:{Ku_A,A}Ks_AS
B->A:{N_A,N_B}Ku_A
A->B:{N_B}Ku_B
Ans: a
a. A->B:{N_A,A}Ku_B
b. B->A:{N_B,A}Ku_B
c. B->S:{N_S,B}Ku_A
d. A->B:{N_B,B}Ku_A

Question 2:
A->S:A,B,N_A
S->A:{N_A,B,Ks_AB,{Ks_AB,A}Ks_BS}Ks_AS
A->B:{Ks_AB,A}Ks_BS
B->A:{N_B}Ks_AB
A->B:{N_B}Ks_AB
Ans: d
a. B->S:{Ks_BS,A}Ks_AB
b. A->B:{Ks_AS,S}Ks_AS
c. A->S:{Ks_AS,B}Ks_AS
d. A->B:{Ks_AB,A}Ks_BS

Question 3:
A->B:A
B->A:N_B
A->B:{N_B}Ks_AS
B->S:{A,{N_B}Ks_AS}Ks_BS
S->B:{N_B}Ks_BS
Ans: c
a. B->S:{B,{N_A}K_BS}K_AS
b. S->B:{A,{N_B}K_BS}K_AS
c. B->S:{A,{N_B}K_AS}K_BS
d. A->S:{B,{N_A}K_AS}K_BS

Question 4:
A->B:A
B->A:N_B
A->B:{N_B}Ks_AS
B->S:{A,{N_B}Ks_AS}Ks_BS
S->B:{A,N_B}Ks_BS
Ans: a
a. S->B:{A, N_B}Ks_BS
b. B->S:{A, N_B}Ks_AS
c. S->B:{B, N_A}Ks_AS
d. A->B:{B, N_A}Ks_BS

Fig. 16. Four identification questions with choices and answers.

procedure. For each subject, our experiment environment first randomly divides the four protocols into two sets, one for the text-based approach and the other for the Lego-based approach, and randomly determines the sequences of protocols in each set. The order of the two approaches is also randomly chosen. Our experiment environment automatically uses the first approach on the first protocol set, and the second approach on the second set.

For each protocol, the system first enters a memorization phase, which allows a subject to study the protocol as long as he or she needs. Then, when the subjects indicate that they are ready, the system shows them one portion of the protocol (with three incorrect alternatives) in the same form that they have been viewing (digital or traditional) and asks them to identify the message that appears in the full protocol previously displayed. This leads to the response phase. After the subjects choose their answers and click the "next" button, our system displays the next question and repeats the same procedure until the experiment is finished. During the experiment, our system automatically records the accuracy, memorization duration, and response duration for each subject and each question.

Experiment Results and Analysis: Figs. 17 (left) and 18 show the statistical results of this experiment. We calculate the averages and standard deviations of the accuracy, memorization duration, and response duration, respectively.

From the results, we can see that the memorization and response durations for these two approaches are similar. The Lego-based approach attracts the attention just a little bit longer than the text-based approach (3.3 seconds). The p-value from t-test is 0.74 showing that this is not significantly different.

The response duration of the Lego-based approach is 9.2 seconds shorter than that of the text-based approach, indicating that the Lego-based approach may be easier for subjects to identify the missing primitives. However, the p-value from t-test is 0.1 showing that this difference is not significant.

The average accuracy of the Lego-based approach is much higher than that of the text-based approach. We think that the low accuracy of the text-based approach shows that the subjects have some difficulties in using the traditional method to identify the missing primitives. Since they have been familiar with the text-based approach throughout the semester, this may reflect some obstacles they have during the course. The absolute accuracy value of the Lego-based approach is also low, but its response duration is shorter and the accuracy is much higher than the text-based approach. The p-value from t-test is 0.03 showing that they are significantly different. We think that this result demonstrates one advantage of the Lego-based approach over the traditional text-based approach.

Experiment 1	Text-based Approach		Lego-based Approach		P-Value
	AVG	SD	AVG	SD	
Accuracy	14.71	29.39	41.17	31.8	0.03
Memorization duration	25.715	31.46	29.028	22.135	0.74
Response duration	45.925	24.828	36.762	16.803	0.1

Experiment 2	Text-based Approach		Lego-based Approach		P-Value
	AVG	SD	AVG	SD	
Accuracy	65.88	26.23	62.35	21.07	0.48
Memorization duration	17.88	9.125	17.452	7.04	0.77
Response duration	10.924	6.082	13.906	5.678	0.1

Fig. 17. The results of experiment 1 (left) and experiment 2 (right).

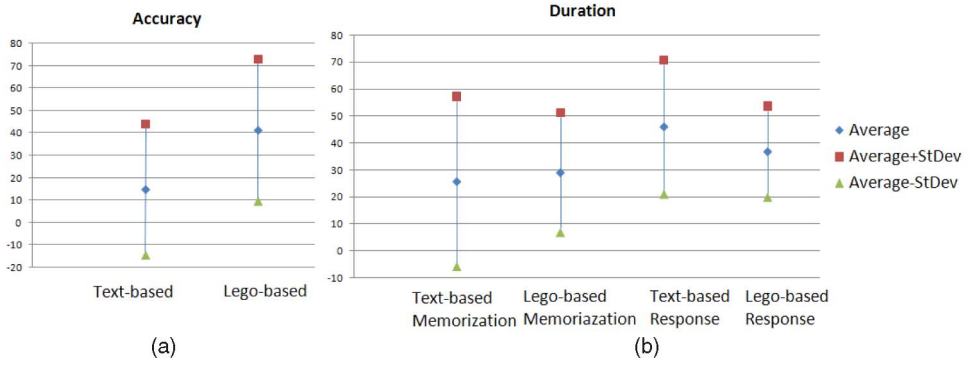


Fig. 18. Data analysis results for experiment 1. (a) shows the accuracy results and (b) shows the memorization and response duration results.

S->A: {B,Ks_AB,N_A,N_B}Ks_AS, {A,Ks_AB!}Ks_BS
Ans: d
a. B
b. A
c. N_A
d. Ks_AB

B->S:B,N_B, {A!,N_A}Ks_BS
Ans: b
a. N_A
b. A
c. B
d. N_B

B->S: {A,B, {A,B,N_B}Ks_AS}Ks_BS!
Ans: a
a. Ks_BS
b. Ks_AS
c. Ks_AB
d. A

S->A:N_B, {B,Ks_AB,N_A}Ks_AS, {A,Ks_AB,N_B!}Ks_BS
Ans: d
a. N_A
b. B
c. A
d. N_B

A->B: {A,Ks_AB!}Ks_BS, {N_B}Ks_AB
Ans: c
a. A
b. N_B
c. Ks_AB
d. Ks_BS

A->B: {Ks_AB,N_B,A}Ks_BS!
Ans: c
a. Ks_AB
b. B
c. Ks_AS
d. Ks_BS

A->B: {A,B,Ks_AB!,N_B}Ks_BS, {N_B}Ks_AB
Ans: a
a. Ks_AB
b. Ks_BS
c. N_B
d. Ks_AS

S->A: {N_A!,B,Ks_AB, {Ks_AB,N_B,A}Ks_BS}Ks_AS
Ans: b
a. Ks_AB
b. N_A
c. N_B
d. Ks_BS

S->A: {N_A,B!,Ks_AB, {Ks_AB,A}Ks_BS}Ks_AS
Ans: c
a. N_B
b. N_A
c. B
d. Ks_BS

B->A: {A,B!,N_A,Ks_AB}Ks_AS, {N_A}Ks_AB,N_B
Ans: b
a. Ks_AS
b. B
c. A
d. N_B

Fig. 19. Ten memorization questions with choices and answers. The primitives followed by a "!" will be replaced by a "?" in the experiments.

6.2.2 Experiment 2: Protocol Primitive Memorization

Since visual-based approaches might be used to strengthen user memory, we design this experiment to evaluate whether or not the Lego-based approach can help subjects remember the primitives in a protocol better than the text-based approach.

Apparatus and Subjects: The same as experiment 1.

Materials: We have selected 10 messages with different kinds of primitives appearing in general security protocols. They are neither too long nor too short. The average number of primitives in these messages is six. Fig. 19 shows the 10 messages, corresponding multiple choice questions, and their answers.

Procedure: The same as experiment 1, our experiment environment randomly divides all the messages into two sets, one for the text-based approach and the other for the Lego-based approach, and randomly determines the question sequences in each set. The order of the two approaches is also randomly determined.

The experiment procedure for each message is similar to the procedure for each protocol in experiment 1. After subjects study a message and click the question button, our experiment environment replaces one primitive in the message with a "?" mark, and displays four choices in the same format as the message. Fig. 11 shows one example of a message in the experiment. The subjects are then asked to identify which primitive has appeared in the previous message. This procedure is repeated until all the questions have been answered. During the experiment, we record the accuracy, memorization duration and response duration for each subject and each question.

Experiment Results and Analysis: Figs. 17b and 20 show the statistical results of this experiment. We calculate the averages and standard deviations of the accuracy, memorization duration, and response duration, respectively.

All the results, including the accuracy, memorization duration, and response duration of these two approaches are similar. The p-values from the t-test also show that they are not significantly different. The small difference between these two approaches may come from the fact that the text-based approach has been used to teach these subjects throughout the semester, while the Lego-based approach is only briefly introduced before the experiment.

Combining the results from our survey and two user studies, we think that the Lego-based approach obviously offers more meaningful and interesting information for students to observe, especially on the relationships among primitives and protocols. Suitable usages of such visual information may lead to direct benefits for students to learn and apply security protocols.

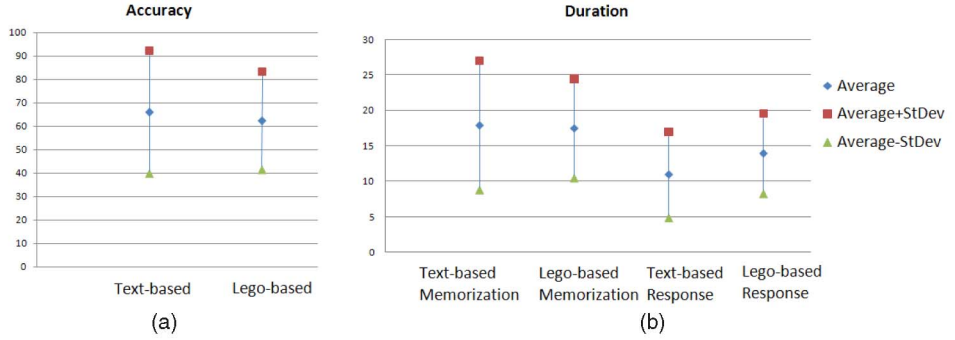


Fig. 20. Data analysis results for experiment 2. (a) shows the accuracy results and (b) shows the memorization and response duration results.

7 Discussion

The main strengths of the Lego-based approach are twofold: attract the interests of students and improve the understanding of security protocols. First, it is essential to the success of information assurance education that we can attract and retain the interests of students. Both the survey and user studies in our evaluation demonstrate that the combination of 3D digital Legos and the text-based approach is the best solution for students to accept. We also emphasize this strength by providing several features to our Lego system, including the flexibility to change primitive designs and the 3D interaction methods that simulate real-life Lego experiences.

Second, the Lego toys promote children to recognize individual shapes and the matching relationships among different blocks. Similarly, our approach constructs digital Legos to help students identify individual security primitives and improve their understanding of the relationships among primitives and protocols. Our user studies evaluate two important aspects, primitive identification and memorization, since they are directly related to our objectives. A good understanding of the relationships among security primitives and protocols cannot be separated from the understanding of individual primitives. During our development process, we explore different designs of primitives, such as the shapes of key words and similar shapes from objects in real-life, to help students link the protocol contents to the shapes of the Lego blocks. The evaluation results demonstrate significantly better primitive identification performance of our Lego-based approach compared to the traditional text-based approach. We believe that once students are familiar with the primitive pieces, better recognition can lead to better memorization of protocol details, and thereby improving the understanding of security protocols. We plan to design more user studies to evaluate other aspects of protocol understanding in the future.

In addition to these impacts, our approach also has the potential to help students understand the linkage between the protocol design and its vulnerabilities. Here we use the man-in-the-middle attack as an example to illustrate the potential. A security protocol is vulnerable to the man-in-the-middle attack when the receiver cannot verify the authenticity and integrity of a message. For example, when A sends its identity and public key in plain-text to B, an attacker on the path can switch A's public key with its own public key. Under this attack, any messages that B intends to send to A can be read by the attacker. We have integrated our digital Lego system with the knowledge model for security protocols [24] to illustrate these attacks. As shown in Fig. 8, for every entity, both its initial knowledge when the protocol starts and the latest knowledge as the protocol proceeds, are shown on the left bottom panel. Therefore, we can combine the content of a message and the latest knowledge of its receiver to identify the components that the receiver cannot verify or authenticate. These components are then labeled in a special color to show that an attacker could have changed their values and a man-in-the-middle attack might exist. Note that this functionality is not dependent on any specific protocols. In fact, we have adopted this technique in our undergraduate level security course to allow the students to understand and compare the man-in-the-middle attacks and type flaw attacks on the key exchange protocols such as Diffie-Hellman and Needham-Schroeder public key protocols.

8 Conclusion and Future Work

To improve the information assurance education, we have developed a digital Lego system for demonstrating and practicing important security concepts. We carefully design our digital Lego sets to provide a generic representation of security protocols. Our approach applies the pedagogical methods learned from toy construction sets by treating security primitives as Lego pieces and protocols as construction results. With our digital Lego sets, we have developed a prototype system and supporting instructional materials. We have also designed and performed evaluations to assess this Lego-based approach and found encouraging results and feedback.

In the future, we plan to introduce our digital Lego approach and course materials gradually into the introductory level security courses. We have collected a list of security protocols that are widely adopted in information assurance education. We will apply interactive visualization techniques to develop supporting functions and integrate them into a more comprehensive experiment environment. We plan to publish our course materials and Lego system online to share with other researchers and educators. We will also continue to perform formal user studies to gather data from larger groups and evaluate the effectiveness of the Lego-based approach on aiding students to understand security protocols. The results of the user studies will be used to improve our Lego-based approach, so that security knowledge can be introduced to a broader population.

Acknowledgments

The authors thank the editors and reviewers for their valuable comments. This research was supported by DOE DE-FG02-06ER25733, NSF 0633150, and NSF 0754592.

References

[1] V. Pothamsetty, "Where Security Education Is Lacking," *Proc. Second Ann. Conf. Information Security Curriculum Development,* pp. 54-58, 2005.
[2] J. Weinberg, G. Engel, K. Gu, C. Karacal, S. Smith, W. White, and X. Yu, "A Multidisciplinary Model for Using Robotics in Engineering Education," *Proc. Am. Soc. Eng. Education Ann. Conf.,* 2001.
[3] P. Wyeth and H. Purchase, "Using Developmental Theories to Inform the Design of Technology for Children," *Proc. Conf. Interaction Design and Children,* pp. 93-100, 2003.
[4] B. Harley, *Constructional Toys.* Shire, 1990.
[5] G. Cross, *Kids' Stuff.* Harvard Univ., 1997.
[6] M. Eisenberg, A. Eisenberg, M. Gross, K. Kaowthumrong, N. Lee, and W. Lovett, "Computationally-Enhanced Construction Kits for Children: Prototype and Principles," *Proc. Int'l Conf. Learning Science,* pp. 79-85, 2002.
[7] Y. Kitamura et al., "Real-Time 3D Interaction with Activecube," *Proc. Conf. Human Factors in Computing Systems (CHI '01),* pp. 355-356, 2001.
[8] M. Resnick et al., "Programmable Bricks: Toys to Think With," *IBM Systems J.,* vol. 35, no. 3, pp. 443-452, 1996.

[9] J. Weinberg and X. Yu, "Robotics in Education: Low-Cost Platforms for Teaching Integrated Systems," *IEEE Robotics and Automation Magazine,* vol. 10, no. 2, pp. 4-6, June 2003.
[10] J. Maloney, L. Burd, Y. Kafai, N. Rusk, B. Silverman, and M. Resnick, "Scratch: A Sneak Preview," *Proc. Int'l Conf. Creating, Connecting, and Collaborating through Computing,* pp. 104-109, 2004.
[11] F. Martin, B. Mikhak, M. Resnick, B. Silverman, and R. Berg, "To Mindstorms and Beyond: Evolution of a Construction Kit for Magical Machines," *Robots for Kids: Exploring New Technologies for Learning,* pp. 9-33, Morgan Kauffman, 2000.
[12] A. Howe, "The Ultimate Construction Toy: Applying Kit-of-Parts Theory to Habitat and Vehicle Design," *Proc. Aerospace Architecture Symp.,* 2002.
[13] C. Coulston and R. Ford, "Teaching Functional Decomposition for the Design of Electrical and Computer Systems," *Proc. 34th Ann. Conf. IEEE Frontiers in Education (FIE '04),* 2004.
[14] J. Millen and V. Shmatikov, "Constraint Solving for Bounded-Process Cryptographic Protocol Analysis," *Proc. ACM Conf. Computer and Comm. Security (CCS '01),* pp. 166-175, 2001.
[15] C. Cremers, "Compositionality of Security Protocols: A Research Agenda," *Electronic Notes in Theoretical Computer Science,* vol. 142, no. 3, pp. 99-110, 2006.
[16] W. Wang, A. Lu, L. Yu, and Z. Li, "A Digital Lego Set and Exercises for Teaching Security Protocols," *Proc. 12th Colloquium for Information System Security Education (CISSE '08),* pp. 26-33, 2008.
[17] C. Burger and K. Rothermel, "A Framework to Support Teaching in Distributed Systems," *ACM J. Educational Resources in Computing,* vol. 1, no. 1, Mar. 2001.
[18] E. Saul and A. Hutchison, "A Graphical Environment for the Facilitation of Logic-Based Security Protocol Analysis," *South African Computer J.,* vol. 21, pp. 26-30, 1998.
[19] D. Schweitzer, L. Baird, M. Collins, W. Brown, and M. Sherman, "Grasp: A Visualization Tool for Teaching Security Protocols," *Proc. 10th Colloquium for Information System Security Education (CISSE '06),* 2006.
[20] N. Elmqvist, "Protoviz: A Simple Security Protocol Visualization," technical report, Univ. of Gothenburg, 2004.
[21] D. Dolev and A. Yao, "On the Security of Public Key Protocols," *IEEE Trans. Information Theory,* vol. 29, no. 2, pp. 198-208, Mar. 1983.
[22] P. Cignoni et al., "Meshlab," http://meshlab.sourceforge.net, 2010.
[23] F. Jacquemard et al., "Security Protocols Open Repository," http://www.lsv.ens-cachan.fr/Software/spore/index.html, 2010.
[24] Z. Li and W. Wang, "Using Deductive Knowledge to Improve Cryptographic Protocol Verification," *Proc. IEEE Military Comm. Conf. (MILCOM '09),* pp. 1-7, Oct. 2009.

Li Yu received the BE degree in computer science and technology from Zhejiang University, China. He is currently a third year PhD student at the University of North Carolina at Charlotte. His research interests include scientific visualization and computer graphics. He is a student member of the IEEE.

Lane Harrison received the BS degree in computer science from the University of North Carolina at Charlotte, where he is currently working toward the PhD degree with a focus on visualization and visual analytics. He has been active in K12 and undergraduate computing outreach for several years. He is a student member of the IEEE.

Aidong Lu received the bachelor's and master's degrees in computer science from Tsinghua University, and the PhD degree in electrical and computer engineering from Purdue University in 1999, 2001, and 2005, respectively. She is now an assistant professor at the University of North Carolina at Charlotte. Her research interests are in developing effective visualization approaches to improve visual communications in real-life applications and education. She is a recipient of Department of Energy (DOE) Early Career Principal Investigator Award in 2006. She is a member of the IEEE.

Zhiwei Li received the BS degree in electrical engineering from Southeast University, China, in 2003, and the MS degree in software engineering from San Yat-sen University, China, in 2006. He is currently working toward the PhD degree at the University of North Carolina at Charlotte. His research interests include security protocol analysis, formal methods, and visualization. He is a student member of the IEEE.

Weichao Wang received the PhD degree in computer science from Purdue University in 2005. He is currently an assistant professor at the Department of Software and Information Systems, University of North Carolina at Charlotte. His research interests are in designing protocols and mechanisms to secure pervasive systems, especially the resource-restraint networks. He is also interested in developing new techniques to improve computer education. He is a member of the IEEE, the ACM, and the ASEE.