An Advance Visual Model for Animating Behavior of Cryptographic Protocols

Mabroka Ali Mayouf Maeref[1*], Fatma Alghali[2], Khadija Abied[2]

[1] Sebha University, Faculty of Science, Department of Computer Science, P. O. Box 18758 Sebha, Libya, Libyan.
[2] Sebha University of Libya, Sebha, Libya.

* Corresponding author. Tel.: 00218-925132935; email: roka.mayouf@yahoo.com
Manuscript submitted February 13, 2015; accepted July 5, 2015.
doi: 10.17706/jcp.10.5.336-346

Abstract: Visual form description benefits from the ability of visualization to provide precise and clear description of object behavior especially if the visual form is extracted from the real world. It provides clear definition of object and the behavior of that object. Although the current descriptions of cryptographic protocol components and operations use a different visual representation, the cryptographic protocols behaviors are not actually reflected. This characteristic is required and included within our proposed visual model. The model uses visual form and scenario-based approach for describing cryptographic protocol behavior and thus increasing the ability to describe more complicated protocol in a simple and easy way.

Key words: Animation, cryptographic protocols, interactive tool, visualization.

1. Introduction

Cryptographic protocols (CPs) mostly combine both theory and practice [1], [2]. These cause protocol complexity describing and understanding. Therefore, separating the mathematical part from the protocol behavior should provide feeling of how the protocol works, thus increasing the ability to describe and to gain confidence in reflecting more complicated information about CPs, as well as to generate interest to know about other more complex protocol concepts.

Several researchers realized the use of visual model and animation techniques to reflect the explanation of the learning objectives and their benefits [3]-[11]. Protocol animation visually shows how the protocol really works and behaves in practice. This feature of animation of cryptographic protocol components and its operations is required to provide the ability of connecting the animated object with its consistent image and behavior. Although different CPs descriptions are used in literatures especially within interactive visualization tools, there is a lack of consideration about these descriptions and their ability for reflecting meaning. According of our knowledge, the possibility of improvement and development of CPs descriptions isn't within attention.

This paper covers the most used descriptions of CPs and proposes an advance visual model as alternative description for better CPs concepts reflecting and understanding. The following section describes the aspects of cryptographic protocol description and most related works to this paper whereas Section 3 explains the proposed model. A discussion of this paper is explained in Section 4 and the conclusion is provided in Section 5.

2. Aspects of Cryptographic Protocol Description

The existing models of cryptographic protocols' descriptions that have been used in literatures are varies. Early literatures includes the formal description of authentication protocols by Burrows *et al.* [12] and Clark and Jacob [13]. These literatures provide a comprehensive explanation of the conventional and formalized notations of authentication protocols. They also explain the problems of the conventional notation of authentication protocols and the importance of the transformation of protocol messages into a logical formula for protocol analysis and verification.

The latest literatures include the descriptions of security protocols especially cryptographic protocols (CPs). Interactive tools such as *Game tool* [14], *ProtoViz* [15], *GRASP* [16], *Kerberos tool* [17], *GRACE* [18], and *CrypTool* [19] use different description model for describing CPs. Based on these literatures, the description might be categorized as follow: Visual model and Text-based model.

The visual model is further divided into two forms: the graphical representation form, and the physical representation form. Table 1 lists the models that were used by the most interactive visualization tools and the following subsections describe each of the models listed in this Table.

Table 1. The Models of Cryptographic Protocol Description

Tool name	The models of cryptographic protocol description
Game tool	Visual model (physical representation)
protoViz	Text-based (formal notation) and visual model (physical representation)
GRASP	Text-based (natural language description) and visual model (graphical illustration MSC)
Kerberos tool	Text-based (natural language description) and visual model (physical representation)
GRACE	Text-based (natural language description) and visual model (physical representation)
CrypTool	Text-based (natural language description) and graphical illustration

2.1. Visual Model

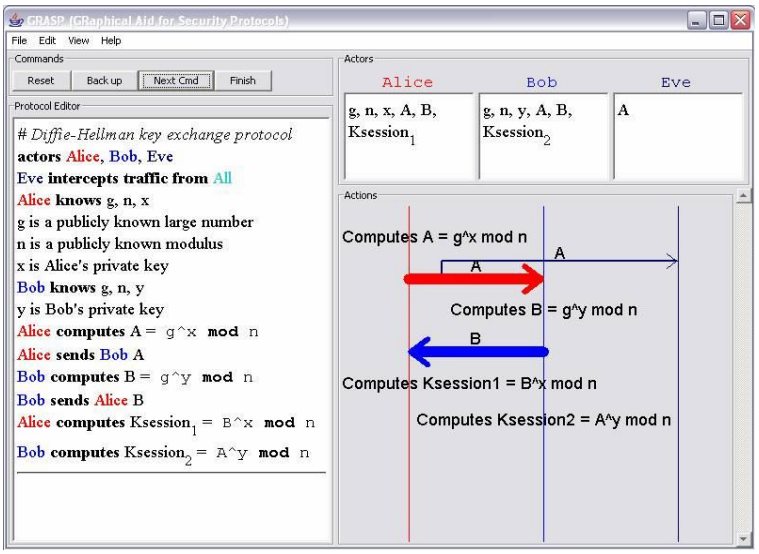

Fig. 1. GRASP tool uses MSCs.

From the literature, we have found that the visual model has been used by all of the five listed tools. GRASP and CrypTool tool use a graphical representation as a visual form while the others use the physical representation form. These physical representations differ from one tool to another. Both the graphical and physical representation forms are discussed in more details as follows:

- Graphical representation form: it uses vertical lines, arrows, information, and values. The common graphical languages used in the field of protocol description are *SDL* (Specification and Description

Language) [20] and MSCs (Message Sequence Charts) [21]. GRASP tool uses MSCs as a graphical illustration for describing cryptographic protocol as shown in Fig. 1.

MSCs can be described as a chart that captures a scenario in which a principal A sends *a message1* to principal B. The principal B in turn sends a reply message to principal A as *a message2*. The vertical lines represent the life-lines of the processes taking part in the scenario. The time is assumed to flow downwards along each life-line. The directed arrows going across the life-lines represent the causal link from a send event (the source of the arrow) to the corresponding receive event (the target of the arrow), with the label on the arrow denoting the message being transmitted. Fig. 2 shows a simple MSC.

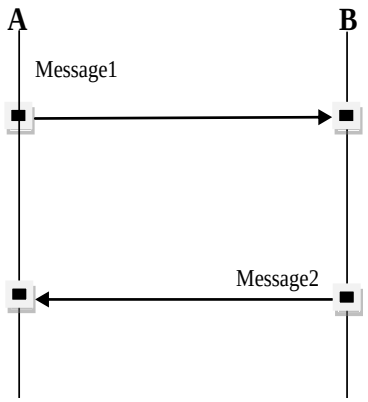

Fig. 2. A simple MSC.

The traditional illustration of cryptographic protocol using MSCs is shown as a vertical timeline of message passing, identifying known information and values. If more information is added to the protocol, the illustration will be more complex. A lot of entities, step descriptions and information will be included within the illustration which cause complicated and difficult description of the protocol. Fig. 3 shows the complexity of the migration protocol illustration using MSCs [22].

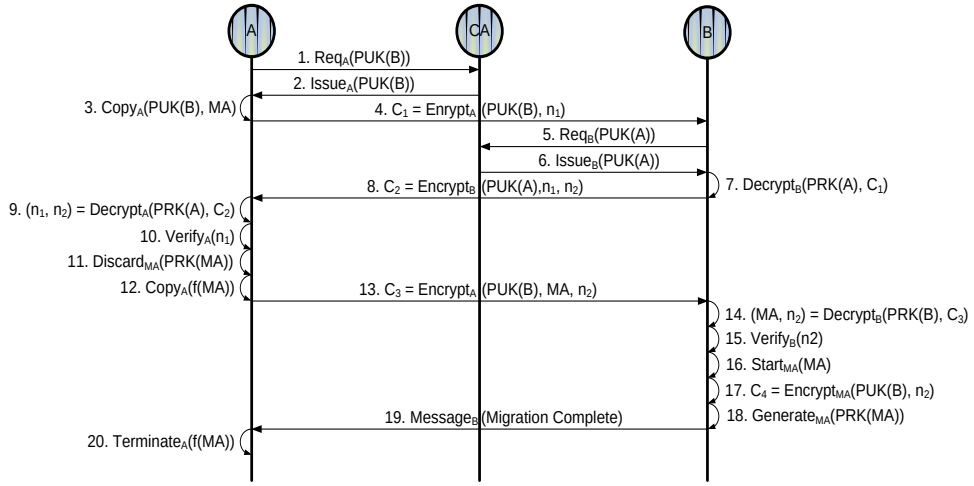

Fig. 3. Steps of migration protocol.

MSCs present, in an intuitive way, how processes interact through message passing. They also constitute an attractive visual formalism that is widely used to capture system requirements during the early design stages in domains such as telecommunication software [23].

MSCs and SDL languages are more concerned with the description and explanation of the formal protocol specification for protocol analysis and validation and that makes the understanding of protocol behavior

difficult [24]-[26].

Cryptool is a freeware software tool with graphical user interface for applying and analysing cryptographic algorithms. It contains nearly all state of the art crypto algorithms with "playful" introduction to modern and classical cryptography.

CrypTool also used graphical representation as sets of boxes and arrows. Boxes describe the input, output and cryptographic algorithm. Arrows shows how data are transferred from one box to the others. The tool also used text-based description to explain the subjects in more details. Fig. 4 shows the main menu and graphical illustration of Caesar Cipher in CrypTool.

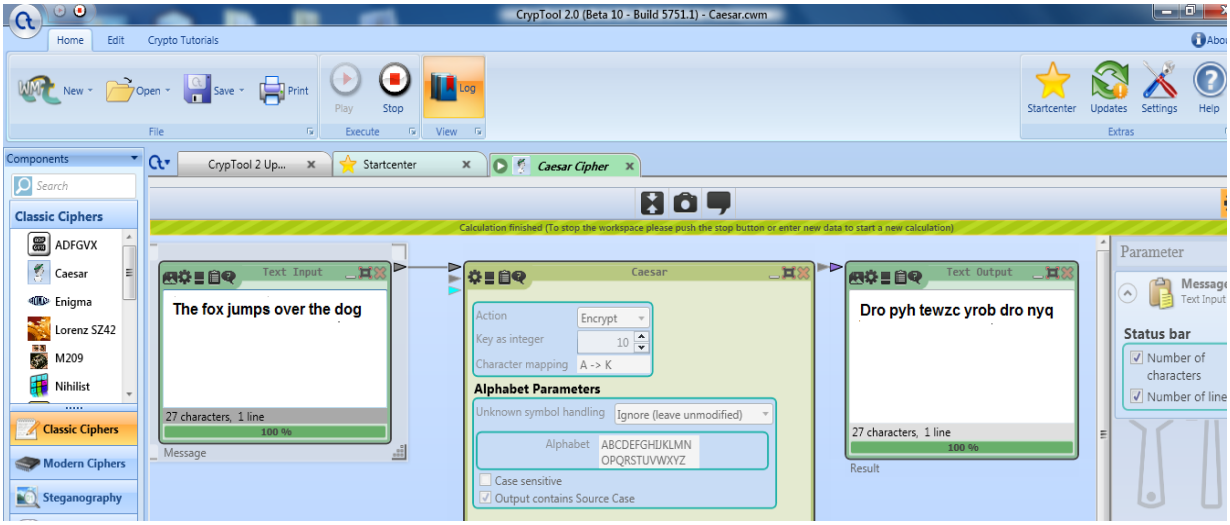

Fig. 4. CrypTool main menu and graphical illustration of Caesar cipher.

- Physical representation form: it means representing an object by a picture or image from the real world. This kind of representation can be a composition of MSC's graphical illustration and simple visual images to provide a visual and intuitive opportunity for understanding the concepts and complex cryptographic protocols. Some cryptography books, [1] and [2], provide these kind of visual figures (refer to Fig. 5).

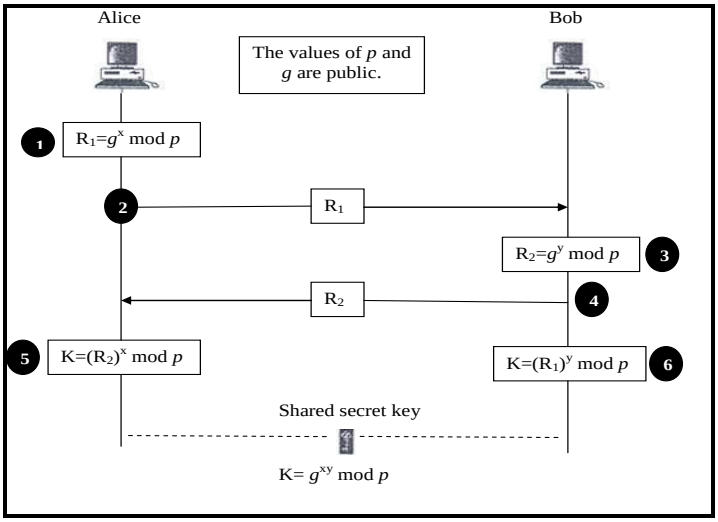

Fig. 5. Diffie-Hellman method.

These figures without animation are rigid, lifeless and less attractive. For these reasons, it should be brought to life through animation to show the real behavior and functions of cryptographic protocols. It

should also be included with interactivity through active learning tools that are full of excitement and attractiveness.

As described in Table 1, Game tool, *ProtoViz*, *Kerberos* and *GRACE* use physical representation forms to describe the cryptographic protocol components. This representation ranges from schematic pictures (*ProtoViz*) to simple realistic pictures (*Kerberos* and *GRACE*). In a meta-analysis of learning through instructional animation versus static pictures by Hoffler and Leutner [27], the authors found that learning from a high-realism animation is more beneficial than learning from a low-realism animation. However, describing cryptographic protocols through a high level of realism is more concern for this paper but still an open question for the previous study.

Looking at the *ProtoViz* tool's physical representation, *ProtoViz* uses schematic pictures, simple drawing pictures and symbols to represent the cryptographic protocol as shown in Fig. 6. ProtoViz tool focuses on the simple design and analysis of cryptographic protocol specifications than behavioral understanding by accepting the protocol input as a formal description transferring it to a simple visual form.

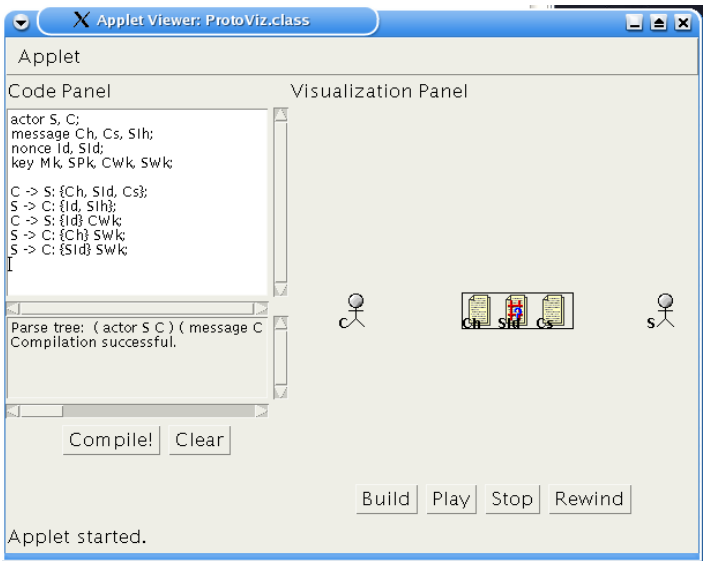

Fig. 6. ProtoViz visual model.

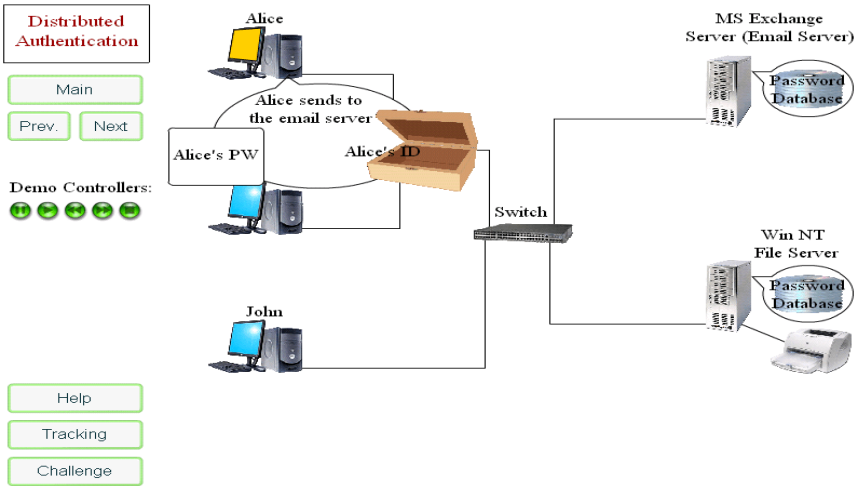

Fig. 7. Kerberos tool visual model.

Kerberos tool uses realistic pictures to represent the Kerberos protocol. It uses real keys, boxes, wires, laptops and devices to represent protocol components and behavior. The *Kerberos* tool focuses on a single special protocol only. It could not be customized to represent other protocols. Fig. 7 shows the physical

representation of the Kerberos protocol.

Looking at *GRACE* tool's physical representation, it uses simple images, data transfer, arrows, and colors to facilitate the understanding of cryptographic protocols as described in Fig. 8.

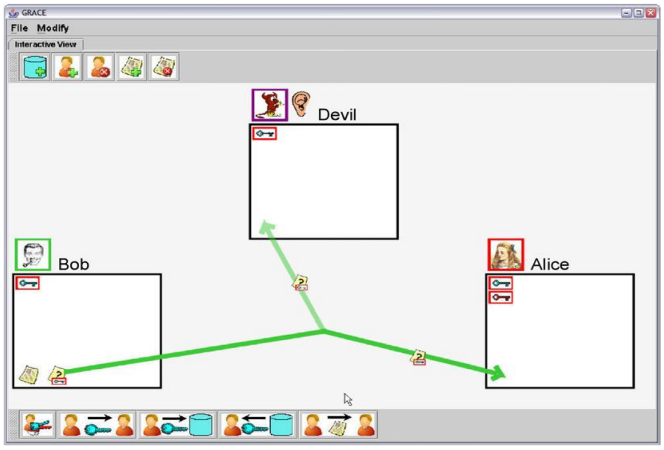

Fig. 8. GRACE tool visual model.

GRACE uses simple images to represent the cryptographic protocol components such as human being pictures, keys and papers as a document. Although cryptographic protocols have different kind of keys, *GRACE* uses the same key to represent all kinds of keys but with different colors to distinguish between them. *GRACE* tool does not mention the properties of the keys, such as lock and unlock, to describe the protocol behavior. It describes cryptographic operations in a simple manner such as the encryption/decryption operations, which are represented by a document paper with/without a question mark. The power of using the physical representation from the real world lies in the use of their properties and qualities in order to describe the behavior of cryptographic protocols. For example, using the keys to lock and unlock the message to explain the encryption/decryption operations.

2.2. Text-Based Description

From the literature we have found that the current text-based descriptions for describing cryptographic protocols are natural language [16]-[18] and formal notation [15]. In text-based description, the words and sentences of natural language are used together with some symbols, to describe the cryptographic protocols. For more clarity, natural language and formal notation text-based descriptions are described briefly as follows:

- Natural language description: This description describes the cryptographic protocol steps using natural language sentences and symbols. Natural language is a language that can be readable and understandable by human beings. In fact, describing the protocol steps using text form (natural language texts) is the first step to define the protocol specifications [1], [2], [28]-[31]. In particular cases, mathematic symbols are involved in the text if computations operations are needed. The natural language description is simple and easy to understand, but it increases error-prone keyboard input by writing words and command lines. It also makes it difficult to capture all these words and commands if more new protocols have been added.
- Formal notation: it's used by formal language as a standard notation with rules and grammar such as a context-free-grammar. Formal notation of cryptographic protocol specification is used for describing and analyzing CPs. Examples of formal notation are idealized form [12], [13] and Derivation form (BAN logic) [12]. An idealized form of protocols facilitates deriving and generating a useful logical form for a protocol. There are guidelines to explain the transformation of particular protocol steps into an

idealized form. Assumptions and logical formulae are written and attached to the statements of the protocol together with assertions about the state of the system after each statement. Finally, the logical postulates are applied to the assumptions and the assertions in order to discover the beliefs held by the principles in the protocol and thus producing a derivation form (BAN logic) of the potocol. However, this notation is quite complex. The user should understand the semantics and grammar of this notation in order to define and specify the protocol.

3. Advance Visual Model

Advance visual model provides strong visual components to represent the specifications of cryptographic protocol such as an actor, message, key, document and data transfer. These visual components used to recreate a wide variety of scenarios in order to animate the behavior of cryptographic protocols. Each specification represented in this model had a suitable visual image and associated with a set of operations. More details are in the following subsections.

3.1. The Actors

An actor represents the person who implements a protocol. Six important actors are found in protocol. Actor named Alice should initiate all the protocols and any of the other actors may be selected to respond. To help distinguish between the actors and their processes, a unique color is associated with each one. This model visualizes each actor by using different images with a text label showing the name of that actor as shown in Table 2. All resources owned by the actor are saved in a special file related to that actor. Each actor has a unique color and all keys owned by that actor have the same color to distinguish between actors and their resources.

Table 2. The Protocol Actors

Name	Description	Visual Form
Alice	First actor in all protocols	
Bob	Second actor in all protocols	
Carol	Third actor in the three- and four party of protocols	
Dave	Fourth actor in the four party of protocols	
Intruder	Malicious active attacker	
Trent	Trusted arbitrator	

Table 3. The Key Visual Model

Key	Visual Form
Public key	
Private key	
Secret key	

3.2. The Keys

Actor uses cryptographic keys to perform cryptographic operations. Three types of keys are included in this model; private key, public key, and secret key. This model visualizes each key as a door key with different style keys as shown in Table 3. The keys' colors are the same as the actor's color who owns the

keys except the secret key. The secret key is always black because nothing is known about this key.

3.3. The Document

This model visualizes document as a paper. It can be used to record the actor's ID (actor identity) or the message that has to be sent to "or received from" the other actor. Each document has a creator and a set of processes. This information saved into a file to keep track of it. This model visualizes a document in different states as shown in Fig. 9. It visualizes plain document as a piece of paper filled with the text and ID document as a piece of paper filled with the name of the actor's ID and Signed document as a piece of paper with the sender's public key sticker at the bottom of the paper. Any document is inserted in a message box to be sent or received.

Fig. 9. Alice's document.

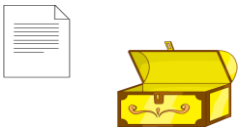

Fig. 10. Document and message box.

3.4. The Message

This model visualizes a message as a box filled with the document as shown in Fig. 10. Keys, cards and any sent item are inserted into the same box.

3.5. An Encryption and Decryption Operations

Using cryptographic keys, this model performs cryptographic operations on the message box such as encryption and decryption operations. If the message box is encrypted using the selected key, then, the message box will be locked by a padlock with the same color of that key. If the message box is decrypted using the selected key, then, the padlock is opened and removed from the message box, if and only if, the color of the key is the same as the color of that padlock. Fig. 11 (a) and (b) visualize the encryption/ decryption operations.

(a) (b)

Fig. 11. (a) Encryption and (b) Decryption of message box using secret key.

3.6. Digital Signature Operation

For signing the document and verifying the signature, the document is signed using the selected key and the signature will be verified using the same selected key. Fig. 12 describes the visualization of the digital signature. If the color of the verifying selected key matches the color of the signing key, then the signature is true, otherwise, the signature is false.

Fig. 12. Signing document using private key.

3.7. Digital Certificate

A certification authority (CA) is a Federal or State organization that binds a public key to an entity and issues a certificate. This model visualizes The CA by a certificate paper signed with the CA private key and stamped by the selected public key as shown in Fig. 13. This model assigns a unique color to CA's private and public keys.

Fig. 13. Visual model of digital certificate.

3.8. Visual Model of Hashing

A cryptographic hash function takes a message of arbitrary length and creates a message digest of fixed length. This model visualizes the hash function as two pinions working together. The message paper is inserted between these two pinions to become a message digest as shown in Fig. 14. In fact, message digest guarantees the integrity of the message. To check the integrity of a message, or document, the hash function run again and a comparison between the new message digest and the previous one is carried out. If both are the same, then the original message has not been changed.

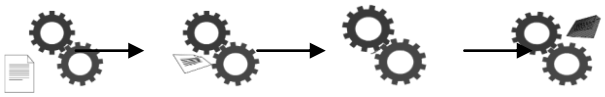

Fig. 14. Visual model of hashing.

4. Discussion

Although the text-based refers to the natural language and symbols' descriptions and the visual model refers to the graphical and physical descriptions, it is strongly realized that, from a human perspective, physical representation is a concrete way to represent the protocol, and a natural language description is more readable than a formal notation. For a machine, it is the other way round where computation can be easily done if the elements can be represented in a formal way.

Our audience is human. Therefore, this proposed model for cryptographic protocols description was considered the human perspective. The model is used for visualizing and animating the cryptographic protocols specifications (components and operations).

5. Conclusion

This paper proposed a visual model that uses more visual representation of the real world instead of using graphs and lines for CPs descriptions. One of the advantages of using a visual representation of our real world is to facilitate the behavioral descriptions by animating the object and behavior with the help of using different images and colors. It also provides the opportunity of rebuilding and defining different scenario using the same object and behavior. Our next target is to include and inject the proposed model into a conceptual design of interactive tool to take place in work.

References

[1] Stallings, W. (2006). *Cryptography and Network Security: Principles and Practices* (4th ed.). Upper Saddle River: Prentice Hall.

[2] Forouzan, B. A. (2008). *Cryptography and Network Security* (1st ed.). New York: McGraw-Hill Higher

Education.

[3] Breimer, E., Cotler, J., & Yoder, R. (2012). Video vs. text for lab instruction and concept learning. *Journal of Computing Sciences in Colleges, 27*, 42-48.

[4] Cooper, S. (2010). The design of Alice. *Transaction on Computing Education, 10,* 1-16.

[5] Hsin, W. (2010). Animations for computer networking protocols. *Journal of Computing Sciences in Colleges, 25,* 245-250.

[6] Rias, R. M., & Zaman, H. B. (2011). Designing multimedia learning application with learning theories: A case study on a computer science subject with 2-D and 3-D animated versions. *Asia-Pacific Forum on Science Learning and Teaching, 12*.

[7] Shapiro, A., & Lee, S. H. (2011). Practical character physics for animators. *Computer Graphics and Applications, 31*, 45-55.

[8] Sohn, E., & Choy, Y. (2012). Sketch-n-Stretch: Sketching animations using cutouts. *Computer Graphics and Applications, 32*, 59-69.

[9] Spanjers, I. A. E., Gog, T. V., Wouters, P., & Merriënboer, J. J. G. V. (2012). Explaining the segmentation effect in learning from animations: The role of pausing and temporal cueing. *Computers & Amp; Education, 59*, 274-280.

[10] Su, Y. (2012). Development of interactive and virtual algorithm animation of c programming language. *Advances in Intelligent and Soft Computing, 169*, 67-72.

[11] Villaverde, K., & Jaramillo, D. (2010). Game design and development course taught with Alice. *Journal of Computing Sciences in Colleges, 26*, 22-29.

[12] Burrows, M., Abadi, M., & Needham, R. (1990). A Logic of authentication. *ACM Transactions on Computer Systems, 8(1),* 18-36.

[13] Clark, J., & Jacob, J. (1997). A survey of authentication protocol literature.

[14] Hamey, L. G. C. (2002). Teaching secure communication protocols using a game representation. *Proceedings of Australasian Computing Education Conference (ACE2003).* Adelaide, Australia.

[15] Elmqvist, N. (2004). *ProtoViz: A Simple Security Protocol Visualization.*

[16] Schweitzer, D., Baird, L., Collins, M., Brown, W., & Sherman, M. (2006). GRASP: A visualization tool for teaching security protocols. *Proceedings of the Tenth Colloquium for Information Systems Security Education* (pp. 1-7). Adelphi, MD.

[17] Yuan, X., Qadah, Y., Xu, J., Yu, H., Archer, R., & Chu, B. (2007). An animated learning tool for Kerberos authentication architecture. *Journal of Computing Sciences in Colleges, 22(6)*, 147–155.

[18] Cattaneo, G., Santis, A. D., & Petrillo, U. F. (2008). Visualization of cryptographic protocols with GRACE. *Journal of Visual Languages and Computing, 19(2),* 258-290.

[19] Esslinger, B. (2010). *The CrypTool Script: Cryptography, Mathematics, and More* (10th ed.). Germany, Frankfurt: Am Main.

[20] ITU-T. (2002). Recommendation Z.100 (08/2002). *Specification and Description Language (SDL)*. Geneva, Switzerland.

[21] ITU-International Telecommunication Union, MSC-2000: ITU-T Recommendation Z.120. (2000). *Message Sequence Chart (MSC)*. Geneva, Switzerland.

[22] Wei, Q., & Patel, A. (2009). A secure and trustworthy framework for mobile agent-based e-marketplace within digital forensics and security protocols. *International Journal of Mobile Computing and Multimedia Communication, 1(3)*, 1-12.

[23] Harel, D., & Thiagarajan, P. S. (2003). Message Sequence Charts.

[24] Genest, B., & Muscholl, A. (2005). Message Sequence Charts: A survey. *Proceedings of the Fifth International Conference on Application of Concurrency to System Design (ACSD'05)*.

[25] Chávez, M. L., & Henríquez, F. R. (2004). SDL specification of a security architecture for WorldFIP. *Proceedings of the 14th International Conference on Electronics, Communications and Computers (CONIELECOMP'04)*.
[26] Turner, K. J. (2002). Protocol animation. *Computer Networks, 40(5),* 595-598.
[27] Hoffler, T. N., & Leutner, D. (2007). Instructional animation versus static pictures: A meta-analysis. Le*arning and Instruction, 17*, 722-738.
[28] Schneier, B. (1996). *Applied Cryptography*. USA: Wiley & Sons, Inc.
[29] Gollmann, D. (2006). *Computer Security* (2nd ed.). John Wiley & Sons, Ltd.
[30] Bishop, D. (2003). *Introduction to Cryptography with Java Applets* (1st ed.). USA: Jones and Bartlett.
[31] Menezes, A. J., Oorschot, P. C. V., & Vanstone, S. A. (2001). *Handbook of Applied Cryptography* (2nd ed.). Boca Raton: CRC Press.

Mabroka A. M. Maeref received her B.Sc. degree in computer science from University of Sebha, Libya in 1991, the M.Sc. degree in computer science from Universiti Sains Malaysia in 2000, and the PhD degree in software engineering from Universiti Kebangsaan Malaysia in 2013.

She is currently working as a lecturer at the Departement of Computer Science, Faculty of Sciences in Sebha University of Libya. Her interests span a wide range of topics in the area of software engineering, networking, security, visual informatic, programming and computer education.

Fatma Abdullah Alghali received a PhD degree in software engineering from University of AL-Neelain SUDAN 2006, a master degree of computer science from Warsaw University of Technology, Poland, 1997, a B.Sc. degree of computer science from Sebha University, Libya, 1991.

She is currently working as an assistant professor in the Computer Science Department of Sebha University, Libya. Her research interest includes software engineering, human computer interactive, e-learning, and cloud computing.

Khadija Abied Ali received the B.S. degree from Sebha University in 1988, the M.Sc. degree from the Technical University of Warsaw in 1997 and the PhD degree in computer science from Czech Technical University in 2008.

She is currently working with the Faculty of Science, Sebha University, Sebha-Libya. Her research interests include database systems, xml, conceptual modelling, query languages, and operating systems.