Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 52 (2015) 121 – 128

Procedia Computer Science

6th International Conference on Ambient Systems, Networks and Technologies, ANT 2015

Anonymous connections based on onion routing: A review and a visualization tool

Abdullah A. AlQahtani, El-Sayed M. El-Alfy*

College of Computer Sciences and Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Abstract

Anonymity of persons and services on the Internet has been an area of growing interest in academia and industry. In this paper, we review work on visualization tools for teaching information security and assurance. We then focus on concepts related to anonymous connections, Onion Routing (OR) protocol, and Tor network. We also present a visualization tool, VISACOR, to demonstrate these concepts. The tool can be used as part of an active and blended instructional strategy for courses related to information security and assurance. We broke down the whole system into several modules that collectively allow the learners to better master the core components, communications, and various modes of operations to achieve anonymity. As a design principle, we aimed to make the tool interactive, comprehensive, and easy to use through animation and user-friendly graphical interface.

Peer-review under responsibility of the Conference Program Chairs
Keywords: Information Security and Assurance Education; Privacy; Anonymous Connections; Onion Routing; Tor Network; Visualization Tools

1. INTRODUCTION

Over the past decade, the interest in information security and assurance has been growing rapidly. With the increasing volume and complexity of the Internet security-related threats and the demand for specialists in this hot area, teaching information security and assurance (ISA) has moved to a new era where full academic programs have been dedicated to it. However, there are several problems and concepts that beginners find difficult to learn and understand. The advances in computer-based teaching, active and blended learning, and visualization technologies have attracted the attention of many researchers and educators to enhance the course delivery in order to engage students more in learning so that they can effectively understand and retain concepts [1–5].

Among the very active areas in information security is private and anonymous communication [6]. While cryptography can be used to scramble the messages to protect against eavesdropping, anonymous connections can additionally hide the identity of the parties involved in the communication. Hence, such connections can protect not only against eavesdropping but also against traffic analysis to make tracing activities back to the source harder. For example, persons surfing the world-wide web or sharing files online may desire to protect their privacy from both the network

* Corresponding author. (On leave from Tanta University, College of Engineering, Egypt.)
E-mail address: alfy@kfupm.edu,sa

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2015.05.040

and outside observers. This censorship-resistant requirement can be crucial to many people using the Internet, especially if their transactions are of special kind[7]. For instance, activists, bloggers and journalists may want to hide their identities from the monitoring authorities. Therefore, their privacy is very important to protect themselves from suspicious activities. Consequently, encryption alone is not enough, but it is required to completely hide the identity as well. Anonymous communication can also be useful in several other situations or work in hostile environments, e.g. hiding an analyst identity while investigating a malware to avoid counter attack, allowing a security expert to bypass network security policies during penetration testing of the network and services, allowing commuters to connect to their home network and services through the Internet when VPN is not available, maintaining privacy in ad-hoc and sensor networks[8].

Several systems, models and protocols have been developed to provide unlinkability or untraceability between the received messages and their true senders and between the sent messages and their true recipients. The Onion Routing (OR) protocol is one of the widely-used techniques to anonymize communications[9]. The user will remain unidentified by the destination or any node in the middle between the source and the destination. Using OR provides the user anonymous connections that are powerful enough to resist eavesdropping and traffic analysis attacks. The first implementation release of The Onion Routing was launched in 2002 and is known as Tor project (`https://www.torproject.org/`)[10]. It uses SOCKS proxy interface and multiple encrypted tunnels to achieve its goals. Formal descriptions and analyses of anonymity and onion routing are presented in[11,12].

Onion routing involves several complex concepts that may generally require excessive effort from the instructors to explain them to beginners in the Information Security and Assurance (ISA) discipline. In order to address this concern, we have developed a tool named VISACOR to VISualize Anonymous Connections based on Onion Routing. Our goal in this paper is to discuss work related to instructional visualization tools for ISA. Moreover, we review anonymity and onion routing based concepts, then we describe the design and implementation of the VISACOR visualization tool. We finally provide some sample cases to demonstrate the tool.

The rest of the paper is organized as follows. Section 2 reviews related work on visualization tools for information security and assurance education. Section 3 provides a brief review of the important background concepts of Onion Routing as a tool for achieving anonymity. Then, we discuss the design and implementation of our visualization tool, VISACOR, in Section 4. After that, we show demonstrations of some cases in Section 5. Finally, Section 6 concludes the paper and highlights future work.

2. Visualization Tools Related Work

Using visualization to help understanding, retention and reasoning about important concepts has been shown to be a very effective instructional strategy[13]. It has been used in several areas in computer science education including networking, algorithms and data structures[14,15].

With the increasing interest in information security and assurance, a number of visualization attempts have been found in the literature. For instance, three educational tools for undergraduate-level security classes are described in[1]. These tools were developed using Macromedia Flash and can be used to demonstrate packet sniffing, Kerberos authentication, and attacks on wireless networks. They have been used and found to be effective in security-related courses at the Department of Computer Science at North Carolina A&T State University.

In information security, denial of service (DoS) attacks are very attractive for hackers to disrupt or ruin the availability of a computing system or service. The attack can be launched in a variety of forms such as SYN flood, fragmented or malformed packets, ICMP flood, smurf amplifier, ping flood, application-level flood, reflective and cooperative distributed denial of service (DDoS). In[16], a visualization-based simulator is described to help students gain knowledge of SYN flood attacks. The tool demonstrates normal network traffic and how a SYN flood attack occurs. It also shows how to prevent SYN flood attacks via a firewall. It was developed in the Department of Computer Science at North Carolina A&T State University and used in COMP 620 Information, Privacy and Security in fall 2010.

Another interactive visualization tool called GRASP is described in[17] to be used in an undergraduate information security class. This tool allows users to interact with arbitrary protocols in a user-controlled stepwise manner.

Firewalls and packet filtering represent a core area in information security. A number of visualization tools have developed to explain and analyze various firewall-related concepts and configurations, e.g.[18–20]. An education tool

for firewall simulation is presented in[21]. This tool can be used to show how the firewall works through interactive sessions.

In the heart of information security and assurance comes cryptology. In 1998, various German Universities and companies have started a project for an open source software to teach cryptography and cryptanalysis; this tool is known as CrypTool (`https://www.cryptool.org/en/`). Another attempt can be found in[22] where the authors designed a tool to help students not only better understand the concepts of cryptography, but also gain significant knowledge of various algorithms and processes of key generation, encryption and decryption. More recently, several visualization technologies for cryptographic concepts, algorithms and protocols are explored in[2]. A group of researchers and collaborators at Michigan Technological University (`http://www.cs.mtu.edu/~shene/NSF-4/`) has been engaged in the development and utilization of a number of other visualization prototypes related to cryptography including DESvisual[23], ECvisual[24], AESvisual[25], RSAvisual[26], VIGvisual[27], and SHAvisual[28]. For instance, ECvisual can be used for teaching ciphers based on elliptic curves over the real field and over a finite field of prime order. In this tool, there are two modes: demo mode and practice mode. The demo mode can be used for classroom presentations and self-study whereas the practice mode allows learners to go through computations by themselves and check the answers using the tool.

3. Understanding OR

As mentioned above, concealing the identity of the Internet users was a concern for many persons involved in special kinds of communications, e.g. bloggers, whistleblowers, journalists, human-rights workers and spies. Even ordinary users may desire to conceal their identities from any surveillance activities. The concept of Onion Routing (OR) has come to provide online anonymity over the public network. It has been developed by the US Naval Research Laboratory in late 1990s. The software implementation of onion routing is given the acronym Tor (a.k.a. The Onion Routing). The Tor network or OR network consists of a set of nodes voluntarily running the Tor software.

3.1. OR Terminology

First, we should be clear about some OR-related terminology that we will use repeatedly in this paper and in OR simulation environment. In onion routing, the sending application is known as the 'originator' or 'initiator' and the receiving application is known as the 'responder'. The initiator communicates to the responder through a sequence of randomly selected intermediate nodes known as 'onion routers'. The sequence of nodes from the initiator to the responder is known as 'circuit' or 'chain'. The first node in a circuit or chain is known as 'entry node' whereas the last node is known as 'exit node' and the other intermediate nodes are known as 'relay nodes'. Each node in the Tor network has a public key and a private key to be used later in encryption/decryption of the messages that come from the OR users (originators or initiators). Information about nodes in the OR network is stored in special nodes called 'directory nodes' or 'directory servers'. In onion routing network, an 'onion' refers to a special type of data structures that encapsulates messages through successive layers of encryption which is analogous to the onion vegetable. Exchanged messages are carried in fixed-length packets known as 'cells'. The Tor network is an overlay network of anonymous connections over the long-standing network connections between onion routers. The 'degree of anonymity' is a quantitative information-theoretic measure based on the concealment guarantee achieved by the Tor network to its users.

3.2. How OR Works?

OR is similar to a virtual private network (VPN) in that both can hide the initiator's true IP address from the intermediate nodes. However, the true IP address is known to the VPN server (the exit point). In contrast, OR has the ability to hide the true identity of the anonymous-connection initiator from the responder (and vice versa) and also from the onion routers on the path from the initiator to the responder (even if they are compromised). Each node on the path from the initiator to the responder is only aware of its upstream node and its downstream node. Messages from the initiator are forwarded from node to node until they reach the exit node where they are finally forwarded to the responder. Once an onion router receives an onion data structure, it peels off the outermost layer, determines the

next hop address, then forwards an embedded onion to the next hop. Note that messages are transmitted from node to node in encrypted form and remain secure as long as they are in the Tor network. But, messages from the exit node to the responder are in plaintext.

The operation of the onion routing network is composed of four phases: (1) network setup, (2) connection setup, (3) data movement, and (4) connection destruction. During the network setup phase, the network topology is predefined and long-standing connections are established between neighboring onion routers. During the second phase, the initiator asks a directory node to return information about a set of nodes (e.g. three nodes) to start a secure and anonymous communication over the Internet. The directory node replies with the requested information including the public key of each node. In the next step, the initiator shares a session key with each of the three nodes. This shared key will be used to encrypt/decrypt messages exchanged between the initiator and each node separately using a symmetric encryption/decryption algorithm. To do so, the initiator uses the Diffie-Hellman (DH) key exchange algorithm. It sends its half of the DH key to the entry node encrypted with the public key of the entry node. Upon receiving this message, the entry node decrypts it and uses its own half of the DH key to form the shared session key. At this point the shared key is only available to the entry node but not to the initiator. To get the shared key at the initiator, the entry node sends its half of the key in a plaintext message to the initiator. The entry node also sends the hash of the shared key to let the initiator verify the the end result. The initiator then adds the received half of the key to its half of the key and checks if the hash of the result equals the hash that comes form the entry node. Now, both nodes have agreed on the same key in a secure manner. Similarly, the initiator should agree on a session key with the second node which is called a relay node. The initiator follows a similar procedure using DH algorithm except that all the communications between the initiator and the relay node should go secure through the entry node. Therefore, the initiator creates a packet containing its half of the DH key encrypted with the public key of the relay node. This part of the packet can only be decrypted by the relay node. The initiator embeds this packet inside another packet that is encrypted with the shared key between the initiator and the entry node. It's like covering the packet by another layer. The outermost layer is encrypted by the shared key between the initiator and the entry node, in this packet the only information is just asking the entry node to forward the encrypted content of this packet to the second node, which is the relay node. This type of packets, called relay packets or cells, is not interpreted by the entry node but only relayed to the next node (a.k.a. relay node).

Relayed cells or packets come with many types. At this point, the goal of the initiator is to extend the circuit to add the next node. The entry node once opens the packet, it finds that the initiator asks to extend the circuit to the relay node. The entry node will change the type of the relay cell to another type called "control cell" this type is used to extend the circuit to the next node. The entry node sends this control cell to the relay node which decrypts it to obtain the initiator's DH half key and adds it to its half key to create a shared key with the initiator. Now, the relay node has shared a key with the initiator without being aware of who or where the initiator is. The relay node replies back to the entry node that it is added to that circuit and provides its half key with the hash of the shared key. The entry node receives this packet and relays it back to the initiator saying that the circuit is now extended.

The initiator can repeat the above steps until it shares a key with each of the three nodes in the circuit. At the end of these steps, the result is sharing a key with each node to use it to encrypt each layer of the cell and form the 'onion', and each node will decrypt only the part that is encrypted with its shared key with the initiator. This is what gives this technique the name onion routing, because the message or the packet is like an onion vegetable consisting of layers (layers of encryption with different keys and each node will decrypt and read its related part only).

The final node in the circuit, a.k.a. the exist node, is what is actually acts on behalf of the initiator. It's the node that gets the command from the initiator to connect to a remote server and the type of data that the initiator wants from that server. The server (responder) sees only the exit node as its client (through a proxy) and the initiator is completely out of the image (or unidentified to the responder).

After the anonymous connection is established, the next phase is to transfer data from the initiator to the responder (forward transfer) and from the responder back to the initiator (backward transfer). This process is in essence similar to the connection-setup phase. When the data transfer phase is complete, the anonymous connection is cleaned up.

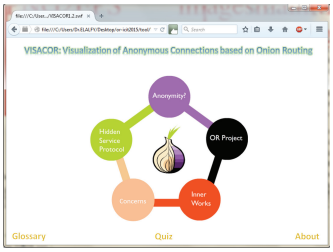


Fig. 1: Main screen of the VISACOR tool.

4. OR Visualization Tool Design and Implementation

From the previous section, we can clearly notice the existence of several concepts and operations that may require a lot of effort to understand and explain to students for the first time. To help students engage and retain these concepts, we developed a visualization tool called VISACOR. In this section, we describe the details of this tool. Users with basic knowledge about information security are expected to benefit from the tool. Those with concepts such as asymmetric and symmetric cryptographic algorithms, key exchange methods, packets routing, client server concepts and exchanging packets in protocols like HTTP will be capable of interacting comfortably with the tool and get the most benefit of it.

The principle design criteria of our visualization tool are as follows. It should be to be easy to use through a user friendly graphical interface. It should be appealing and interactive to engage students in learning. It should use animations to explain concepts and operations related to onion routing. It should use audio-visual effects to be useful for a wide group of users including those with audio or visual impairments. We broke down the tool into the following manageable modules:

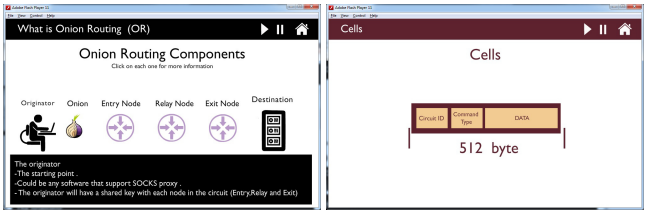
- *About*: provides general information about the visualization tool.
- *Flash Card Glossary*: provides definition for important terminology related to anonymity and onion routing.
- *Anonymity*: explains the basic knowledge about anonymity and onion routing. The various types of messages and components and their roles in achieving anonymous connections.
- *OR Inner Works*: represents the core of the tool. It allows the user to understand the required software, settings and operations to construct circuits, move data forward and backward, and destroy connections.
- *Location-Hidden Services*: demonstrates how services can be anonymized in the Tor environment.
- *OR Related Concerns*: discusses various issues, weaknesses and critiques on anonymity connections and onion routing including security concerns, cryptographic overhead, load balancing and latency performance. It also explains how to measure anonymity quantitatively.
- *Test Understanding*: allows the user to take a quiz to test his understanding of covered concepts and operations.

The tool is aimed to be comprehensive, clear and easy to use. Therefore, we divided the content into scenarios and each scenario is represented in the form of scenes. The tool was developed using Adobe Flash Professional CS6. The main screen of the tool is shown in Figure 1. The user can select any of the main sections in the tool to explore its details. In every scene there is a button to return to the main screen. We use animations and text to explain the concepts and sequences of operations. We also used synchronized audio files generated using text-to-speech (TTS) to help users with visual impairment to benefit from the tool.

5. Demonstration of Sample Cases

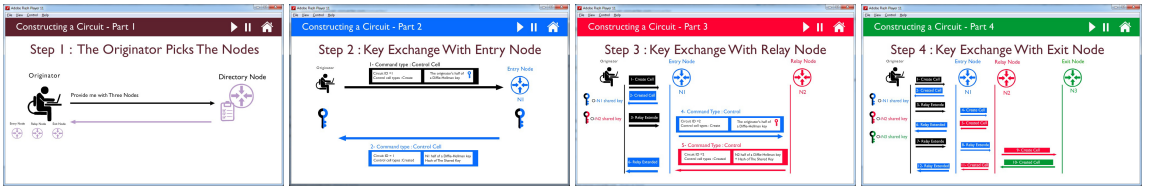
5.1. Anonymity?

In this section of the tool, there is a brief introduction about anonymity and onion routing. It explains to the learner the main components that make up the OR and how they are ordered. The user interacts with the scene by clicking on



(a) (b)

Fig. 2: (a) Main components of Onion Routing, (b) Cell scene.



(a) End of Step 1 (b) End of Step 2 (c) End of Step 3 (d) End of Step 4

Fig. 3: Demonstration of connection setup from the initiator to the exit node.

each component to learn more about its role in the anonymity support system. Concepts covered here are essential for a beginner to understand basic knowledge of OR and continue using the tool. Later, the user can optionally skip this part and explore the rest of the sections. As an example, Figure 2 (a) shows the main components in onion routing where the user can click and learn more about each of them. The user must also understand the type of messages used in OR and the way each packet/cell is organized. Figure 2 (b) shows the cell scene which explains everything related to cells. Some of the cells are called control cells which will be interpreted by the receiving node. Other cells are called relay cells which will not be interpreted by the receiving node, instead they will be forwarded to the next node.

5.2. Inner Works: Connection Setup

This section of the tool provides the core operations and related concepts in onion routing. These concepts include key exchange and encrypting/decrypting messages. The steps in this section are ordered logically to mimic the reality of OR. When interacting with this scene, colors are used to avoid confusion of mixing concepts. Among the important operations is circuit construction or connection setup from the initiator to the exit node as demonstrated in Figure 3. The details are explained as follows:

5.2.1. The initiator picks nodes

The first step discusses how the initiator picks a set of onion routers (nodes) from a directory node and how the communication process proceeds. In this step, we use animation to visualize the messages transmitted by each party involved in this process. The animation starts by the initiator asking the directory node for a set of nodes, e.g. the initiator wants three nodes. The directory node will reply with the information about the three nodes and how to connect with them using the addresses and the public keys to encrypt the communication among them. The initiator goal at this step is to collect the information about the node to prepare for the next step which is sharing a session key with each node in the circuit.

5.2.2. Key exchange with entry node

This step visualizes the how the initiator agrees on a key with the first node in the circuit (the entry node). With this key, the initiator and the entry node will encrypt and decrypt all communications between them using a symmetric cryptographic algorithm. The user will see how they agree on the key using Diffie-Hellman algorithm to exchange the key in a secure manner.

5.2.3. Key exchange with relay node

After the previous step, the circuit consists of the initiator and the entry node. Now, the initiator wants to extend the circuit to add the next node (the relay node). The initiator has the public key of the relay node and will repeat the same steps used to agree on a key. The communication between the initiator and the relay node will pass through the entry node, but the entry node will not have access to the data that is encrypted using the public key of the relay node. The entry node will see only the information related to relaying or routing of the message. In case there are multiple entry nodes, the process is repeated to extend the circuit to other nodes one by one.

5.2.4. Key exchange with exit node

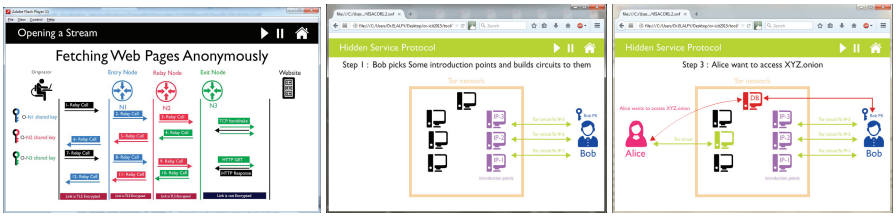
The initiator's goal here is to share a key with the last node in the circuit which is the exit node. This node is important since it's the node that will communicate to the server on behalf of the initiator.

5.3. Inner Works: Data Transfer

This scene is about anonymous communication with a particular server (responder). Once a circuit is ready and all components share keys with each others, each node cannot recognize who is the initiator. Each node only knows about the node before and after (through a proxy). The relay node cannot understand if itself is an entry node. The entry node can not recognize if the initiator is any other node in the circuit. In other words, the communication between the initiator and the responder will be anonymous through the other nodes. Neither the responder nor the nodes in the circuit know the true identity of the initiator. Figure 4 (a) demonstrates the data transfer scenario where the initiator is trying to fetch a web page from a web server anonymously.

5.4. Location Hidden Services

Location-hidden services or responder anonymity uses the concept of Rendezvous Points (RP) to allow servers such as web servers to offer services without revealing their true IP addresses. The VISACOR tool explains concepts related to this topic and how it can be achieved in Tor network. As an example, Figure 4 (b) and (c) show a sample of scenes for the demonstration of location-hidden service advertisement and registration.



(a) (b) (c)

Fig. 4: Demonstration of: (a) data transfer scenario, (b) and (c) sample of scenes for location-hidden services.

6. Conclusion

This paper introduced a new tool that visualizes concepts related to anonymous connections via onion routing, which can be a useful supplement to actively engage learners in information security and assurance related courses. We discussed the importance of visualization tools in education and reviewed some of the recent related works. After that, we dived more into the details and concepts of the onion routing to see what services it can provide and what mechanisms it uses to achieve its goals. We also discussed security and performance issues related to onion routing. As future work, we plan to extend the tool to add more interactions such as allowing the user to take various roles and become part of the normal operation or to act maliciously.

Acknowledgment

The authors would like to thank King Fahd University of Petroleum and Minerals (KFUPM) for the support during this work. The second author would like also to acknowledge the support provided by King Abdulaziz City for Science and Technology (KACST) through the Science & Technology Unit at King Fahd University of Petroleum and Minerals under project No. 11-INF1658-04 as part of the National Science, Technology and Innovation Plan.

References

1. Yuan, X., Vega, P., Qadah, Y., Archer, R., Yu, H., Xu, J. Visualization tools for teaching computer security. *ACM Transactions on Computing Education (TOCE)* 2010;**9**(4):20.
2. Simms, X., Chi, H. Enhancing cryptography education via visualization tools. In: *Proc. 49th Annual Southeast Regional Conference*, 2011.
3. Schweitzer, D., Gibson, D., Collins, M. Active learning in the security classroom. In: *Proc. 42nd Hawaii International Conf. on System Sciences, HICSS'09*, 2009.
4. Yu, H., Williams, K., Xu, J., Yuan, X., Chu, B., Kang, B., et al. Interactive simulation tools for information assurance education. In: *Proc. 2nd Annual Conf. on Education in Information Security*, 2009.
5. Schweitzer, D., Brown, W. Using visualization to teach security. *Journal of Computing Sciences in Colleges* 2009;**24**(5):143–150.
6. Edman, M., Yener, B. On anonymity in an electronic society: A survey of anonymous communication systems. *ACM Computing Surveys* 2009;**42**(1).
7. Häyry, M. Academic freedom, public reactions, and anonymity. *Bioethics* 2014;**28**(4):170–173.
8. Kamat, P., Zhang, Y., Trappe, W., Ozturk, C. Enhancing source-location privacy in sensor network routing. In: *Proc. 25th IEEE International Conf. on Distributed Computing Systems*, 2005.
9. Reed, M.G., Syverson, P.F., Goldschlag, D.M. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications* 1998;**16**(4):482–494.
10. Dingledine, R., Mathewson, N., Syverson, P. Tor: The second-generation onion router. Tech. Rep.; Naval Research Lab Washington DC; 2004.
11. Mauw, S., Verschuren, J.H., de Vink, E.P. A formalization of anonymity and onion routing. In: *Computer Security*, 2004, p. 109–124.
12. Camenisch, J., Lysyanskaya, A. A formal treatment of onion routing. In: *Advances in Cryptology*, 2005, p. 169–187.
13. Hansen, S., Narayanan, N., Hegarty, M. Designing educationally effective algorithm visualizations. *Journal of Visual Languages & Computing* 2002;**13**(3):291–317.
14. Schweitzer, D., Brown, W. Interactive visualization for the active learning classroom. In: *ACM SIGCSE Bulletin*; vol. 39, 2007, p. 208–212.
15. Lahdenmäki, M. *Software Visualization for Teaching Network Protocols*. Ph.D. thesis; School of Science and Tech, Aalto University; 2010.
16. Terry Jr, T., Yu, H., Williams, K., Yuan, X., Chu, B. A visualization based simulator for SYN flood attacks. In: *IMAGAPP/IVAPP*. 2011, .
17. Schweitzer, D., Baird, L., Collins, M., Brown, W., Sherman, M. Grasp: A visualization tool for teaching security protocols. In: *Proc. 10th Colloquium for Information Systems Security Education*, 2006.
18. Warner, J., Musielewicz, D., Masters, G.P., Verett, T., Winchester, R., Fulton, S. Network firewall visualization in the classroom. *Journal of Computing Sciences in Colleges* 2010;**26**(2):88–96.
19. Mansmann, F., Göbel, T., Cheswick, W. Visual analysis of complex firewall configurations. In: *Proc. 9th International Symposium on Visualization for Cyber Security*, 2012, p. 1–8.
20. Lee, C.P., Trost, J., Gibbs, N., Beyah, R., Copeland, J.A. Visual firewall: real-time network security monitor. In: *IEEE Workshop on Visualization for Computer Security, (VizSEC'05)*, 2005, p. 129–136.
21. Williams, K., Yu, H. An interactive firewall simulator for information assurance education. In: *Proc. International Conf. of Information Security and Internet Engineering*, 2011.
22. Abuzaid, A., Yuan, X., Yu, H., Chu, B. The design and implementation of a cryptographic education tool. In: *Proc. 3rd International Conf. on Computer Supported Education*,2011, p. 193–198.
23. Tao, J., Ma, J., Mayo, J., Shene, C.K., Keranen, M. DESvisual: A visualization tool for the DES cipher. *Journal of Computing Sciences in Colleges* 2011;**27**(1):81–89.
24. Tao, J., Ma, J., Keranen, M., Mayo, J., Shene, C.K. ECvisual: a visualization tool for elliptic curve based ciphers. In: *Proc.43rd ACM Technical Symposium on Computer Science Education*, 2012, p. 571–576.
25. Ma, J., Tao, J., Keranen, M., Shene, C.W. AESvisual: A visualization tool for the AES cipher, 2014; `http://www.cs.mtu.edu/~shene/NSF-4/`.
26. Tao, J., Ma, J., Keranen, M., Mayo, J., Shene, C.K., Wang, C. RSAvisual: a visualization tool for the RSA cipher. In: *Proc. 45th ACM Technical Symposium on Computer science education*, 2014, p. 635–640.
27. Li, C., Ma, J., Tao, J., Keranen, M., Shene, C.W. VIGvisual: A visualization tool for the vigenère cipher, 2014; `http://www.cs.mtu.edu/~shene/NSF-4/`.
28. Ma, J., Tao, J., Keranen, M., Shene, C.W. SHAvisual: A visualization tool for secure hash algorithm, 2014; `http://www.cs.mtu.edu/~shene/NSF-4/`.