# Simulation of PKI-Enabled Communication for Identity Management Using CyberCIEGE

C.E. Irvine and M.F.Thompson
Naval Postgraduate School
Monterey, CA USA
irvine@nps.edu mfthomps@nps.edu

***Abstract*—CyberCIEGE is a sophisticated network security simulation packaged as a video game and used by educators around the world to enhance information assurance education and training at universities, community colleges, within the DoD, and in other government agencies. The CyberCIEGE game engine was recently expanded to include Public Key Infrastructure (PKI) features including certification authorities, selection of installed roots and cross certification. CyberCIEGE Virtual Private Network (VPN) gateways, VPN clients and email clients were then extended to incorporate the new PKI features. CyberCIEGE PKI abstractions are described in terms of player configuration choices and the consequences of these choices on network management and vulnerabilities. The CyberCIEGE game engine modifications include modeling of chains of trust and risks of cross certification schemes. The benefits of these enhancements include coherent integration of identity management technologies, ranging from the human interface through to the supporting distributed infrastructure, into scenarios. Benefits also include support for recent new scenarios focused on the PKI infrastructure, identity management, or both; and the ability to tie both identity management and PKI to concepts of identification, authentication, provenance, and access control.**

***Keywords-network cyber security; identity management; information assurance; educational video game***

## I. Introduction

CyberCIEGE is a video game that is used to enhance computer network security education and training through constructive resource management techniques such as those employed in the Tycoon© games [1][2]. In the CyberCIEGE world, players spend virtual money to build, operate and defend networks, and can watch the consequences of their choices, while under attack.

The extensible nature of CyberCIEGE allows its developers and the larger CyberCIEGE community to serve a wide range of teaching objectives and desired participant experiences. At the simple end of the spectrum, scenarios can be constructed to make students aware of the importance of strong passwords and the dangers of email attachments [3]. More complex scenarios teach students how to configure systems and networks, for example how to set up router filtering to support various policy objectives.

In the CyberCIEGE interactive environment, players[1] are guided through a series of scenarios that highlight various cyber security education and training objectives. These include such topics as user passwords, e-mail attachments, antivirus and identity theft. More advanced scenarios cover significant aspects of network management and defense including the use of link encryptors, network filters, VPNs, access control mechanisms, and identity devices such as card readers and biometric scanners. Players make tradeoffs and prioritization decisions as they are challenged to maintain a balance between budget, productivity, and security. Players must keep the virtual world's personnel happy (e.g., by providing Internet access), while protecting assets from vandals and professional attacks.

CyberCIEGE is used by a range of DoD and other government agencies to enhance information assurance education, training and awareness. It is also used in dozens of universities and community colleges. CyberCIEGE was created by the Center for Information Systems Security Studies and Research (CISR) at NPS, and Rivermind, Inc., of San Mateo, California. CyberCIEGE is available at no cost to agencies of the US Government.

Our paper begins with an overview of CyberCIEGE's components. Then we describe the CyberCIEGE network simulation, and that is followed by a discussion of game engine extensions made to represent identity management, PKI functions and PKI-enabled applications. These extensions are designed to help students understand issues related to the management of the identity of users and the identity of data. This work included configurable VPNs, email encryption and authentication, and support for simulated PKI functions within the game.

## II. Basic CyberCIEGE Components

CyberCIEGE consists of several elements: a domain-specific *simulation engine* and *scenario definition language*; a

These CyberCIEGE extensions were sponsored by the Biometrics Task Force.

[1] We use the word "player" to refer to students who interact with the game simulation. Here, the term "student" is used rather broadly, and can include: traditional students in K-12, colleges, and universities; employees requiring awareness regarding cyber security issues; technicians who require specific training, etc. We use the word "user" to refer to the virtual characters within the game.

*scenario development tool*, and a *video-enhanced encyclopedia*. [3]

The game is extensible in that new CyberCIEGE scenarios tailored to specific audiences and topics are easily created. The scenario definition language expresses security-related risk management trade-offs to be developed for different scenarios. The CyberCIEGE simulation engine interprets each scenario as written in the scenario definition language and presents the player with the resulting simulation. Specific player experiences and the consequences of the player choices are a function of the particular scenario.

The game engine and the language that feeds it are rich in cyber security concepts; it is possible to simulate sophisticated environments subject to a variety of threats and vulnerabilities. Substantial support is also provided for relatively brief, scripted training and awareness scenarios, and includes cartoon-like balloon speech by the virtual users, message tickers, multiple choice questions that direct game play, pop-up quizzes and conditional play of video sequences.

Instructors may assess students through logs produced as a result of player activity. Triggers within the scenario cause output to be appended to each log, where a variety of status indicators may be recorded. A separate log is maintained for each player, thus allowing the instructor to track the progress of individual students.

## III. Overview of the Network Security Simulation

The CyberCIEGE game engine assesses the virtual users' ability to achieve their goals [4]. These goals are defined in terms of read and write accesses to assets. Similarly, the game engine assesses the virtual attackers' ability to read and write assets when driven by motives the scenario designer associates with those assets. Both of these functions rely on a simulation of the network topology created by the player within constraints imposed by the scenario designer using the Scenario Development Tool [5].

### *A. User and Attacker Access to Information Assets*

The game engine manages the network topology as a collection of computers and network devices interconnected via one or more networks. The simulation represent information as "assets", which exist on computers. Virtual users and attackers access assets either directly by interacting with the computer that contains the asset, or indirectly via one or more network connections. Additionally, attackers can compromise assets via direct access to networks (i.e., via a wire tap). And attackers might attempt to bribe authorized users to disclose or modify assets.

Players interconnect CyberCIEGE networks using routers, VPN gateways and link encryptors. Workstations and servers may be connected to multiple networks. Components include simulated operating system functions that provide access control mechanisms, supporting policies (e.g., authentication and identification) and application policies such as a router filtering of network connections. The game abstracts away hubs and network switches such that a single named network can connect many computers and network devices.

Direct access to computers and networks is constrained by simulated physical security mechanisms (e.g., walls, guards, and locks.) Logical access to computers is constrained by authentication and identification mechanisms including authentication servers, card readers, biometric scanners, and password policies. Once logical access to a computer is achieved, logical access to assets is further constrained by operating system-based access control mechanisms (e.g., an ACL).

The simulation assesses network topology, logical access controls and physical access controls to evaluate whether virtual users can achieve their goals to access assets and whether attackers can compromise the assets.

### *B. Simulation Fidelity*

The fidelity of the simulation is designed to be high enough for players to make meaningful choices with respect to deploying network security countermeasures, but not so high as to overwhelm the player with administrative minutia. The purpose of the game is not to train players to deploy and configure specific network products. Rather, CyberCIEGE is intended to illustrate abstract functions of technical protection mechanisms and configuration-related vulnerabilities. For example, an attack might occur because a particular firewall port is left open and a specific software service is not patched. The game does not include abstractions to represent network addressing (and thus not Network Address Translation) and it is not intended to illustrate denial of service attacks other than relatively abstract attacks that take place against individual components.

Instead of identifying problems with real-world networks and training network operations personnel to recognize and respond to specific network attacks, a driving philosophy behind CyberCIEGE is that when the most serious network attacks occur, there is often little or nothing to see. Somehow your competitor gets your secrets. Or your enemy adds features to your logistics management system. (Clearly, if an organization discovers that it has been successfully attacked, a postmortem analysis can sometimes reveal useful information about the attack.)

### *C. Assessing User Goals*

When a virtual user attempts to achieve a goal, the game engine assesses the network topology and determines if the user has physical and logical access to a source workstation such that one of the following is true:

- The asset is on the source workstation
- The asset is on a computer on a network shared with the source workstation
- The asset is on a computer on a network reachable from a network containing the source workstation via one or more network devices (e.g., routers)

The network topology processing uses a brute force enumeration of paths that might lead from the source workstation to the computer that contains the asset. With the exception of email services, computers do not act as gateways to other connected networks when processing user goals. Given a suitable path between the source workstation and the asset required to achieve a user goal, the game engine associates the asset with each network segment that the asset would traverse for use in subsequent wiretap attack processing.

### *D. Assessing Attacker Access to Assets*

When virtual attackers attempt to compromise assets, the attack engine employs a similar assessment of the network topology and the access control mechanisms. However, the attacker can use computers as gateways to other networks, and attackers may also supply their own computer and connect it directly to accessible networks, including the simulated Internet. This computer can then indirectly access the computer containing the targeted asset, or the computer can wiretap the network and collect or alter assets that transit the network as the result of some user achieving a goal as described above. Attackers may also employ malicious software, e.g., a Trojan horse that alters the asset (in the case of integrity-driven motives), or sends the contents of the asset to a network accessible by the attacker.

Malicious software appears on computers and network devices in a variety of ways, including poor configuration management and virtual users who compulsively open email attachments. Within the simulation, malicious software also arrives via unpatched network services (e.g., web servers) and whose services are not blocked by network filters.

### *E. Network Filters*

As noted above, goals are defined in terms of reading or writing assets. Goals may also be defined in terms of the types of software required to achieve the goal. For example, a goal requiring read-write access to an email asset might also require an email transport application. The scenario designer may designate the goal software as *filtered,* which may further constrain user (or attacker) access to assets via networks due to network filter settings. For example, a router's network filter might be configured to block email transport application traffic. This would prevent a virtual user from achieving an email goal via that router, and it would prevent a moderately motivated attack from exploiting flaws in the email transport application.

### *F. Assurance of Mechanism vs Attacker Motive*

When determining the success of attacks, attacker motive is compared against the strength of the simulated logical and physical mechanisms that potentially protect the asset from compromise. For example, a network filter will not prevent an attack if it is hosted on a router whose operating system is weaker than the attacker's motive. Within the simulation, the strength of protection mechanisms is represented as an "assurance" value that the scenario designer associates with individual operating systems and applications. To protect assets, the protection mechanisms must also be properly configured (e.g., if an asset's ACL permits "public" access, it may be vulnerable regardless of the strength of the operating system.)

### *G. Link Encryptors Protect Network Communication*

Simulated link encryptors come in two forms: those that represent manually loaded keys and those that represent software-based key distribution. Players configure these devices by selecting which network connections are encrypted and which are not. Manually loaded key systems require the player to select the same key on both devices, and the engine tracks key life such that scenario designers can simulate stale keys, e.g., by triggering player feedback and penalties if the key life exceeds a designer-specified value. Software-based key management systems don't experience stale keys, but their assurance is limited by the assurance of the operating system within the platform. With suitable attacker motive, software-based link encryptors are vulnerable to two kinds of simulated attacks:

- Wiretapping in which the data is transmitted in a key known by the attacker;
- Indirect network attacks in which the attacker uses a subverted link encryptor as a gateway along the path toward the computer hosting the targeted asset.

### *H. Original CyberCIEGE VPN Gateways*

Initially, CyberCIEGE was designed so that VPN gateways only protected Internet traffic, and they supported a single local network connection. Players could not otherwise configure VPN gateways and there were no VPN clients. Traffic between two VPN gateways was assumed to be encrypted and authenticated. VPN gateways were vulnerable to the two attacks just described for software-based link encryptors.

## IV. EXTENDED VPN FEATURES

In support of a comprehensive identity management education program, [6] CyberCIEGE has been extended to better simulate identity management of data. As part of this work, CyberCIEGE VPN gateways were re-implemented so that they could offer protection over any network, not just the Internet. VPN gateways now also support multiple LAN connections, and CyberCIEGE workstations can be configured to use VPN client mechanisms. Players may configure VPN mechanisms to selectively protect traffic depending on its source and destination, simulating the management of IPSEC security associations [7]. For example, the player can configure a gateway to encrypt and authenticate all traffic between a local LAN and a remote business partner. Similarly, the same VPN gateway can be configured to permit unprotected traffic between that same local LAN and the remainder of the Internet.

### *A. VPN Connection Profiles*

The game refers to these VPN configurations as "connection profiles". For example, consider the topology represented in Fig. 1. Table 1 illustrates a connection profile within the "Local VPN Gateway" of Fig. 1, the first entry of which results in encryption and authentication all data between Joe's computer and the Branch Server.

Connection profiles may include wildcard values to indicate any matching item. The second entry in Table 1 matches all traffic. Connection profiles are an ordered list, and the game selects the first entry that matches the source and destination of the traffic. Thus, the second entry in Table 1 would cause the local VPN gateway to permit unprotected traffic between Joe's workstation and other destinations on the Internet.

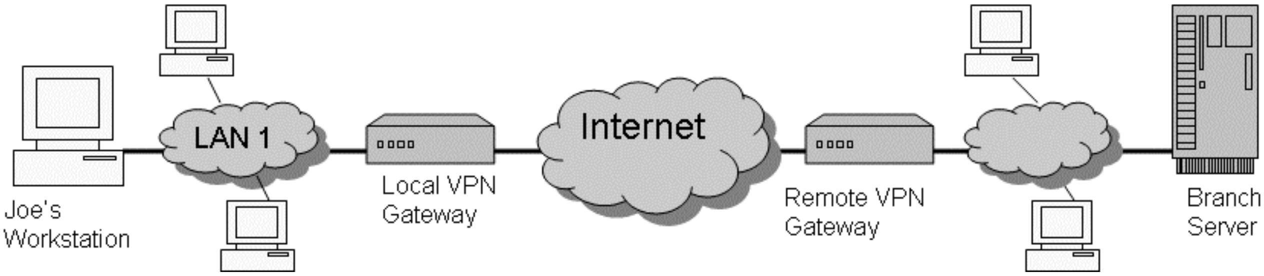

Figure 1. Example VPN Topology

TABLE I. LOCAL VPN GATEWAY CONNECTION PROFILE

| Local Network | Local Host | Remote Network Domain | Remote Host | Protection |
|---|---|---|---|---|
| LAN 1 | Joe's Computer | Our Branch Office | Branch Server | Encryption & Authentication |
| * | * | * | * | None |

VPN mechanisms occur in pairs (i.e., two gateways or a gateway and a VPN client). The remote VPN gateway in Fig. 1 must have a connection profile that encrypts and authenticates traffic between Joe's workstation and the branch server. Otherwise, the two gateways could not exchange traffic.

### B. *Educational Purpose of Connection Profiles*

When players configure connection profiles, they make decisions regarding the identity management of data. For example, by selecting "authentication" as the protection for data exchanged via a remote VPN gateway, the player establishes the identity of the source of the data. Connection profiles also educate players regarding the limitations of such mechanisms. Consider an example in which a valuable secret asset resides on the branch server in the example topology represented in Fig. 1. The connection profile within the local VPN gateway (per Table 1) ensures that when Joe accesses the remote secret, the traffic is encrypted and authenticated. And the remote VPN gateway may be configured to ensure that such communication only occurs with the specifically identified VPN gateway, (i.e., the one at Joe's site). Note however that by letting Joe access the rest of the Internet via unprotected traffic, the local VPN gateway permits a Trojan horse within Joe's computer to send a copy of the secret asset to an attacker on the Internet.

CyberCIEGE scenarios lead the player to potentially configure such vulnerable systems by giving the users, e.g., Joe, goals to access the remote secret asset as well as surfing the web. Low motive attacks may be countered by ensuring Joe's computer is free of malicious software via good configuration management, antivirus, network filters and procedural policies. Later in the scenario, a higher motive asset is introduced, and the likelihood that this higher motive results in malicious software cannot be countered with mere "good practices." In such a case, the easiest solution for the player is to buy Joe a second workstation and connect a second LAN via which to surf the Internet while reserving the other workstation and LAN for accessing the high value secrets.

### C. *Handling VPN's within Network Routes*

As the game engine performs its enumeration of possible routes between users (or attackers) and assets, it keeps track of open VPN tunnels such that every route segment can be queried with respect to open VPN tunnels (including potentially nested VPN mechanisms). If a route terminates at an attacker's computer with open VPN tunnels, the assurance values of the remote VPN mechanisms are compared with the attacker's motive to compromise the asset. If the route terminates with no open VPN tunnels, then the presence of VPN mechanisms does not hinder access to the asset.

When goals are achieved and the presence of assets on network segments is recorded, the game engine also records properties of VPN tunnels that are open when the asset hits the network segment. Subsequently, during processing of wiretap attacks, the assurance of the VPN mechanisms that open the tunnels and those that close the tunnels are compared against the attacker motives to compromise the assets.

### D. *Identity of a Platform*

Instead of deploying pairs of VPN gateways, the player may choose to configure VPN clients within workstations to communicate with a remote VPN gateway. One game scenario forces the player toward this solution by not providing enough funds to purchase two VPN gateways. The VPN clients are configured very much like the VPN gateways.

With respect to information assurance, one of the most significant distinctions between a VPN gateway and a VPN client is that the former is often more easily locked down into a strictly managed configuration. On the other hand, VPN clients exist within workstations whose configurations change frequently, e.g., through installation of new programs or device drivers. Recently, commodity computing platforms have included a Trusted Platform Module (TPM) that can attest to the identity of the platform, including software loaded on the platform [8]. A TPM can be used to manage the keys for a VPN client such that the keys are not unlocked unless the platform has an expected software configuration and an authenticated user. Loading unauthorized drivers on such a platform would prevent the keys from being used, and thus might prevent the VPN mechanism itself from being subverted.

CyberCIEGE VPN clients can be configured to require a "measured boot" to ensure the platform boots into a known state. The simulated workstation must then also be configured to measure its boot process and attest to the identity of the platform. If the player makes these security-enhancing selections, then attackers must have a commensurately higher motive in order to subvert the simulated VPN client mechanisms.

### *E. Key Management*

When configuring VPN mechanisms, players can choose between the use of symmetric secret keys and PKI. When the player selects the former, costs are incurred to simulate the challenge of physically distributing secret keys, with the highest costs occurring when shared secrets must be established with a remote business partner. Selection of secret keys may also leave a system vulnerable to rouge VPN mechanisms within the enterprise. A scenario designer might define a VPN gateway that happens to use the default shared secret key, but cannot be managed by the player. Unless the player explicitly changes the secret keys within the other VPN mechanism, the rouge gateway can become a conduit via which assets are compromised.

A significant part of the game engine extensions was introduction of PKI features into the game. Players can choose to use a PKI-based key management for their VPN mechanisms instead of shared secrets.

## V. PKI SIMULATION

### *A. PKI-based VPNs*

If a player chooses to manage VPN keys using a PKI, the player must select the root certificates that each PKI mechanism will accept. Scenarios offer players the use of "pay-per-cert" public Certification Authorities (CAs). In addition to costing the player funds for each certificate, this choice can result in spoofing due to bogus certificates issued by the public CA. As an alternative, players can purchase their own CA, which is generally less vulnerable to spoofing, (but does require the player to hire adequate IT support staff.)

When players select installed root certificates, they make choices about who may make representations about the identity of remote VPN mechanisms. Installing the pay-per-cert root is convenient, but players soon discover the CA may not be that careful when it comes to signing certificates[2].

Each PKI-based VPN mechanism must also have an assigned CA, i.e., the CA that has signed the VPN mechanism's associated certificate. Two VPN mechanisms cannot establish a protected tunnel unless they can validate each other's certificates. The game allows players to cross certify another organization's certificate such that users can communicate with remote business partners via a VPN. Thus, the game simulates the validation of certificate chains. Use of cross certification can lead to the player's enterprise being spoofed by bogus certificates issued by the business partner. Within the game, risks introduced by cross certification can be mitigated by configuring VPN connection profiles with a certificate policy. Such a policy can restrict communication to remote VPN mechanisms whose certificates chains are free of cross certified certificates. Thus, the player may configure a VPN gateway to require locally generated certificate chains for sensitive intra-enterprise traffic while allowing cross certified chains when communicating with a business partner.

### *B. PKI Keys within VPN Routes and Attacks*

During attack processing, open VPN tunnels are queried to determine the installed roots within the respective VPN mechanisms, and to determine the CAs that sign each mechanism's associated certificates. The attacker motive is compared against the assurance of the CA. Thus, the strength of a VPN tunnel is limited by the assurance of the VPN platforms as well as the assurance of the mechanisms and procedures used when making representations about identity.

Within the simulation, CAs managed by the enterprise are not subject to subversion and spoofing if they are kept off of networks and the motive is not extreme. Networked CAs, partner CAs and pay-per-cert CAs have assurance levels limited by the assurance of their underlying platforms.

### *C. PKI for Email Encryption and Authentication*

Scenario designers create email assets by defining goals that require email clients and email transport applications [5]. Players configure procedural policies to direct the user to use some combination of encryption and signing for email sent as part of selected user goals. Similarly, the player directs the user to authenticate email received as part of other email goals. Scenario designers can contrive scenarios that force players to make decisions other than use brute force and "turn everything on," In one scenario, a remote business partner will not accept encrypted email because they don't want to interfere with the "deep packet inspection" performed by their intrusion detection system.

The game simulates the use of PKI to perform key management within the game's email clients. Players select installed roots and certification authorities in a manner similar to the configuration of VPN mechanisms. Purposely crafted scenarios lead the player to understand that if secrecy is not required, it may be best to sign and not encrypt an email. For

[2] One game CA offers a guarantee that before they issue a certificate, they collect their fee.

example, since signing an email does not require the signer's email client to validate a certificate, the signer does not require 3rd party roots or cross certification. Additionally, players can elect to configure the client to use smart card readers in place of keys managed by the email client application. Within the game, suitably motivated attacks will compromise the private keys managed by the email client on the workstations, while smart cards do not present such risks, reflecting the fact that private keys never leave the smart card.

In addition to protecting communications over networks, email encryption and signing provides protection of data at rest on the email servers. The potential value of this is illustrated in an email scenario in which the player must operate under a management mandate to outsource the email server to a contractor who is criminally motivated to view the asset. In that scenario, failure to direct virtual users to encrypt their email leads to disclosure of the assets. In that same scenario, a virtual user is compelled to access email from a workstation controlled by the malicious contractor. Failure to deploy smart cards and smart card readers results in the contractor obtaining the user's secret key and disclosing all of the user's encrypted email on the server. The next section will describe vulnerabilities associated with email on the workstations.

### D. *Email Attacks and Network Routes*

Within the simulation, the cryptographic algorithms utilized to encrypt and sign email are assumed to be quite strong. Attacks are focused on the management of keys, and the moments in which email is unencrypted (i.e., while being composed or viewed).

As was described above for PKI-based VPN mechanisms, the network routes keep track of CAs whose roots are used when validating certificates and CAs that sign certificates. The associated assurance is compared with the attacker motives to compromise the emails. As noted earlier, the use of client-managed keys for email signing and decryption introduces a potential for the compromise of the private key. The game engine simulates this vulnerability this by creating a temporary asset to represent the key. The attacker motive to compromise the key is derived from the motive to compromise the email assets. The attack engine then attempts to compromise the key. If smart cards are used, no key asset is created. However the smart cards themselves might become a conduit for the compromise of assets. If the same smart card is used to access email on two different networks (e.g., one sensitive and the other unclassified), the attack engine views the smart card as a network link between the two respective workstations. Malicious software on the workstations would then transfer the content of the asset from the sensitive network to the unclassified network.

Within the simulation, an asset generally exists on one component. Other than tracking network segments traversed to achieve a virtual user's goal, the simulation does not involve the movement or copying of assets between components. When processing attacks on email assets, the game engine creates temporary copies of the asset on the sender's workstation and on the receiver's workstation. These copies are in plain text and are unsigned, regardless of the protection granted asset on the email server. The attack engine is then aimed at these temporary assets. Thus, if the email is composed on a workstation containing a Trojan horse, a suitably motivated attacker might get a copy of the email (assuming the attacker has a network connection to the workstation.)

## VI. Conclusions and Future Work

The addition of PKI features to CyberCIEGE gives players an opportunity to explore the abstract functions of different kinds PKI components. Players learn the role of PKI within network computer security architectures and how different applications make use of PKI to support the management of the identity of sources and destinations of data. Players also learn about potential risks associated with the use of PKI by experimenting and observing cause and effects.

The game engine simulation of Secure Sockets Layer (SSL) connections is currently being re-implemented to include the new PKI functions. This includes the management of server-side certificates as well as optional deployment of client side certificates (including the use of smart cards). As is the case with use of PKI for VPNs and email, players must configure installed roots and certificates to permit users to achieve goals. Players can direct users to require "the little padlock" in their browser when achieving selected goals. As is the case with all CyberCIEGE procedural policy suggestions, the virtual users are more likely to abide by the policy if suitably trained. Companion scenarios will illustrate issues related to self-signed certificates, as well as risks of relying on SSL to protect high value assets.

## References

[1] C.E. Irvine, M.F. Thompson, and K. Allen, "CyberCIEGE: gaming for information assurance", Security & Privacy Magazine, IEEE, May-June 2005, Volume: 3, Issue: 3, page(s): 61- 64, ISSN: 1540-7993

[2] A. Rollings and E. Adams, Fundamentals of Game Design. Prentice Hall, 2006.

[3] B.D. Cone, C.E. Irvine, M.F. Thompson, and T.D. Nguyen, "A video game for cyber security training and awareness", Computers & Security 26 (2007) pp. 63-72

[4] C.E. Irvine, and M.F. Thompson, "Expressing an information security policy within a security simulation game", Proceedings of the Sixth Workshop on Education in Computer Security (WECS6), Naval Postgraduate School, Monterey, California , July 12-16 2004, pp. 43-49

[5] Naval Postgraduate School, The Center for Information Systems Security Studies and Research, "CyberCIEGE Scenario Development Tool User's Guide", , http://cisr.nps.edu/cyberciege/downloads/sdt.pdf. Last accessed 17 April 2010.

[6] Naval Postgraduate School, Identity Management Education Program, http://imep.nps.edu/ . Last accessed: 17 April 2010.

[7] S. Kent, and R. Atkinson, "Security architecture for the internet protocol", RFC 2401, November 1998.

[8] "Trusted Platform Module (TPM) Specifications". Trusted Computing Group.