Formal Verification of a Secure Model for Building E-Learning Systems

Farhan M Al Obisat
Computer and Information Technology Dept
Tafila Technical University
Tafila, Jordan

Hazim S. AlRawashdeh
Departement of Computer Science
Buraydah Private Colleges
Buraydah, Saudi Arabia

***Abstract*—Internet is considered as common medium for E-learning to connect several parties with each other (instructors and students) as they are supposed to be far away from each other. Both wired and wireless networks are used in this learning environment to facilitate mobile access to educational systems. This learning environment requires a secure connection and data exchange. An E-learning model was implemented and evaluated by conducting student's experiments. Before the approach is deployed in the real world a formal verification for the model is completed which shows that unreachability case does not exist. The model in this paper which is concentrated on the security of e-content has successfully validated the model using SPIN Model Checker where no errors were found.**

***Keywords*—Formal verification; SPIN Model Checking; E-content; E-protection; Encryption and Decryption; Security of e-content**

I. Introduction

A formal verification for a secure e-learning system model was designed for implementation by computer centers in universities. Figure 1 indicates a secure model for building e-learning systems [1]. This study formally verifies the suggested model, which presents a wireless system that provides university users with remote access to the database files of students. A new security system is proposed to verify if e-learning application environment has weaknesses and to assess data cryptography at rest and in transit. [1] proposed a system that could validate user input for malicious data. In their proposed system, access switches connect all PCs, and a core switch then connects all wireless devices to the access point and secures them using open virtual private network (openVPN) on the client side with a MAC address. The application servers of students located in the computer center are connected to the core switch. This setup adds to a secure connection. The firewall guards the core switch and the entire network. The university is connected remotely using OpenVPN and Pretty Good Privacy (PGP).

The study applies a SPIN model checker to verify the proposed model. The model is presented as a Unified Modelling Language (UML) state diagram using the ArgoUML CASE tool.

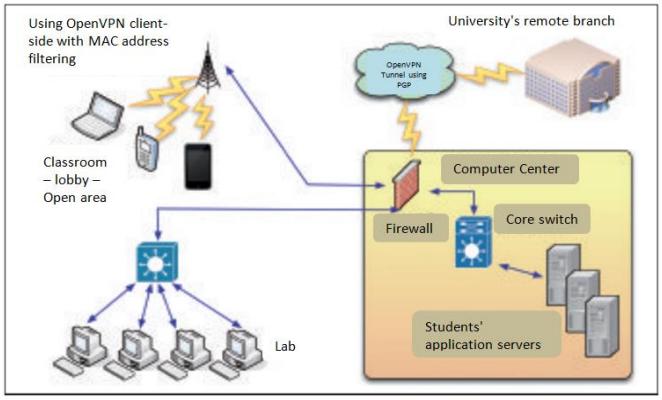

Fig. 1. Secure e-Learning System Model

II. Related Work

A. Formal Verification

A significant area in formal methods is the concept of formal verification. Formal methods include mathematical techniques and tools that are applied for models, specifications, and verification of systems [2]. These factors focus on the formal behavior description of systems, and specifications reflect the degrees of these systems.

In formal verification, the state space of the system utilizes model checkers for determining the specification properties [3]. If these properties are valid, model checkers will return empty files. However, errors will result in a file which is generated by SPIN and called the *TRAIL* file (i.e., output file of the SPIN model checker) or will generate a counter example (i.e., output file of the SMV model checker) to present the process of the violation. Thus, the model checker could present as a series of model states that contain model variables and their values at that state, which aggravate the violations [4].

To check the model system, a finite state machine has to be verified. SPIN verification aims to check violations of safety and liveness factors [5]. Safety properties should prevent errors from occurring. For example, the model should not be exposed to invariants and deadlocks, which would prevent the achievement of possible states.

SPIN inspects a safety property by searching for traces that could direct to an "undesired" element. The lack of trace then satisfies the property [3]. Therefore, the SPIN model checker evaluates assertions that can be utilized between any two statements in the state space [6]. When the model checker determines a calculation that can cause false assertion, the program encounters an error or the assertion is unable to express a correct property [6]. The evaluation is simulated with a true or false expression for a specific statement. If a statement progresses correctly, then the model checker will proceed to the next statement. However, if the statement is unable to progress correctly, then the program will terminate by indicating a trail showing the number of these errors [7].

Linear Tree Logic (LTL) can be applied in several cases to model the time sequence. These models can be translated by SPIN into a never claim, which is then executed together with the finite automaton that represents the Process or Protocol Meta Language (PROMELA) program. LTL may be used to verify certain properties of the system, such as safety, liveliness, and lack of deadlocks [8].

The application of SPIN to formal verification has been commonly used in evaluating security models. For instance, [9] analyzed the security of an approach using behavior-based anti-phishing, and [10] used the SPIN model checker for verifying the security of an anti-phishing model and they found no deadlocks on that model.

This paper evaluates the approach's efficiency by means of formal methods. This research verifies the E-learning model presented in Figure 2 in a formal technique. The verification helps to check whether the model is viable (i.e. un-reachability case does not exist) so as to apply it in the real world. The E-learning model is verified in a formal manner using SPIN model checker. The approach in this paper uses UML state diagram to specify the state model and the state transitions based on the diagram. This UML diagram is translated into PROMELA language code (i.e. the input language of SPIN model checker) so it can be analyzed by SPIN.

B. Secure e-Learning Systems

Researchers have presented the security issues and weaknesses of e-learning systems from various perspectives. [11] elaborated that the security issues of an e-learning schema use four pillars that should be positioned to enhance overall security. These pillars boost e-learning security, present e-learning security policies and procedures, apply e-learning security counter measures, and scrutinize the e-learning security countermeasures.

On the other hand; [12] introduced the security features of e-learning authentication. Using web application for security requires utilizing the SKiP method to provide the same features of SSL. Moreover, using RIPEMD-160 hash function is suggested to provide security and authentication whereas [13] argued that Information Security Management (ISM) is essential to safeguarding the security of the e-learning environment. Combining ISM and information security technology could assure a better security implementation of the e-learning system. This step assures improved results of the successful implementation of security. [1] proposed a model that could deal with securing the e-content of the e-learning system and improve mobile access to several learning systems.

III. MATERIALS AND METHODS

The verification methodology used in this research was successfully used in previous papers by [9] and [10] where this research focuses mainly on the formal verification of the proposed e-learning model using SPIN model checker. In this case, SPIN could be used to verify the model against vulnerabilities based on the system's mechanism that's shown in figure 2. Each part of the model is considered as a different process. These processes are recognized in PROMELA (Process or Protocol Meta Language). PROMELA reads the behavior of the processes that's described in the state diagram in figure 2. These model entities communicate with each other using some global channels. Accordingly, the E-learning model is designed in UML (Unified Modeling Language) state diagram using a case tool called ArgoUML. The generated UML model is then mapped into a PROMELA code using a tool called Hugo/RT which can capture the properties of the model and save it as a PROMELA program. For formal verification process the generated code is combined with some LTL properties that will be used for verifying the timing of the model responses as illustrated in Figure 2.

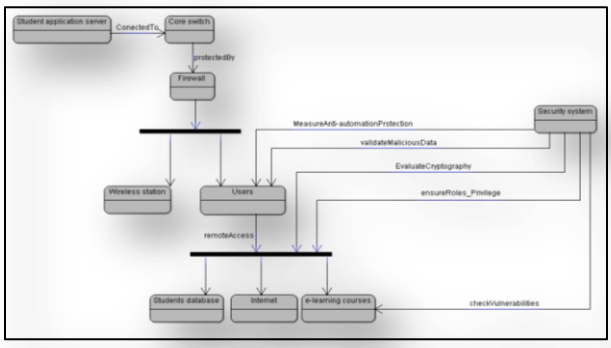

Fig. 2. E-learning Model State Diagram

As stated in Figure 2 student application server is connected to a core switch which is protected by a firewall for the wireless station as well as the users. The security system can work for multiple tasks, i.e. the measurement of the anti-automation and the validation of malicious data of users as well as ensuring the roles and privileges for the database, the internet and the e-learning courses and checking the vulnerabilities for these courses.

IV. RESULTS AND DISCUSSION

The PROMELA code for the e-learning model is translated and written to fit the SPIN model checker. The code is written as follows:

```
1   proctype E-learning (){
2   printf("initiating E-learning  Process...\n");
3   DisplayMenuState:
4   UserInterface!DISPLAY_Options->
```

```
5   printf("E-learning  Process: connected to Core
    switch...\n");
6   printf("E-learning  Process: Protected by
    Firewall...\n");
7   goto Securitysystem;
8   printf("initiating E-remoteaccess  Process...\n");
9   UserInterface!security systems->
10  do:: MeasureAnti-automationProtection
11     :: validateMaliciousData
12  printf("E-learning  Process:
    AccessToDdatabase...\n");
13  printf("E-learning  Process: AccessToInternet...\n");
14  printf("E-learning  Process: AccessToE-
    learningCourses...\n");
15  Od::
16 goto Securitysystem;
17 do:: EvaluateCryptography
18    :: ensureRoles_Privilege
19 printf("E-learning  Process:
   AccessToDdatabase...\n");
20 printf("E-learning  Process: AccessToInternet...\n");
21 printf("E-learning  Process: AccessToE-
    learningCourses...\n");
22 od::
23 goto Securitysystem;
24 do:: checkVulnerabilities
25 printf("E-learning  Process: AccessToE-
   learningCourses...\n
26 od::
27
28   ...
29
30   }
```

After the PROMELA code is written for the model, SPIN verification checks for deadlocks and unexecuted codes in the e-learning model. Figure 3 shows that SPIN does not show any "invalid end state," given that no deadlock is observed in the model. Moreover, the result shows no error and unexecuted codes, and all processes have zero unreached states.

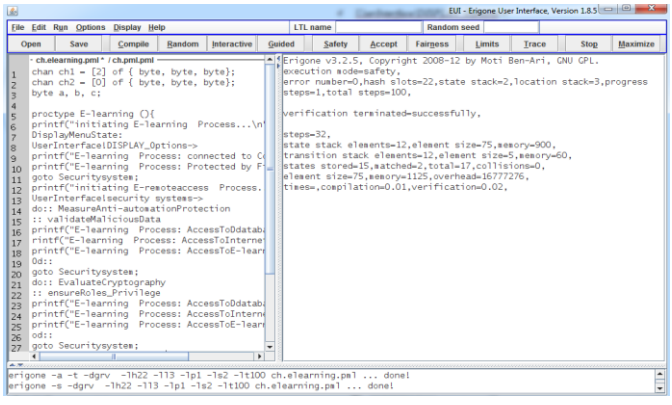

Fig. 3. SPIN Model Checker Result

The states *initiating E-remoteaccess*, *AccessToDatabase*, *AccessToInternet*, and *AccessToE-learningCourses* after *SecuritySystem* have multiple entry points. Figure 3 shows the verification of the properties of these states with the LTL properties. The results indicate that the proposed e-learning model passes these properties during checking. Therefore, the SPIN model checker can validate the model as indicated by the absence of deadlocks or unreachable states.

V. Conclusion and Future Work

This research met the objectives of the study by plotting the model and formally evaluating the proposed e-learning model. Further, the study effectively coded the model using PROMELA and validated the program using the SPIN model checker. The method checked for deadlocks and unreachable states, which did not emerge from the model. Future research could look into proposing another model checker, such as the Symbolic Model Verifier, which could be applied to gain clear results on the veracity of the behavior of the e-learning model.

Acknowledgment

The authors would like to thank Dr. Abdullah Alnajim for his appreciated cooperation in providing with some valuable information on the process of verifying secure systems.

References

[1] Shadi R Masadeh, Nedal Turab, Farhan Obisat, 2012. *A secure model for building e-learning systems. Network Security*, Volume 2012, Issue 1, January 2012, Pages 17-20.

[2] Wing J. M., "*A specifier's introduction to formal methods*," Computer, vol. 23, no. 9, pp. 8–23, 1990.

[3]] M. Machin, F. Dufossé, J. Guiochet, D. Powell, M. Roy and H. Waeselynck, "Model-Checking and Game theory for Synthesis of Safety Rules," *2015 IEEE 16th International Symposium on High Assurance Systems Engineering*, Daytona Beach Shores, FL, 2015, pp. 36-43.

[4] Matthew L. Bolton, Noelia Jimenez, Marinus M. van Paassen and Maite Trujillo "*Automatically Generating Specification Properties From Task Models for the Formal Verification of Human–Automation Interaction*" IEEE Transactions On Human-Machine Systems, vol. 44, No. 5, October 2014R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[5] Manu S. Hegde, Jnanamurthy HK and Sanjay Singh, "*Modelling And Verification Of Extensible Authentication Protocol Using* SPIN *Model Checker*" International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.6, November 2012.

[6] Hegde, M. S., Jnanamurthy, H. K., & Singh, S. (2012). *Modelling and Verification of Extensible Authentication Protocol using SPIN Model Checker*.International Journal of Network Security & Its Applications (IJNSA), 4(6), 81-98.

[7] Samanta, R., Deshmukh, J. V., and Emerson, E. A. (2008, November). *Automatic generation of local repairs for boolean programs. In Formal Methods in Computer-Aided Design*, 2008. FMCAD'08 (pp. 1-10). IEEE.

[8] Michael Huth and Mark Ryan, (2007) Logic in Computer Science, Cambridge.Philippsohn S. "*Trends In Cybercrime — An Overview Of Current Financial Crimes On The Internet*", in Computers & Security, 20 (1), 2001 pp. 53-69.

[9] Abdullah M. Alnajim, "An Evaluation of A Country Based AntiPhishing Approach Using Formal Methods" International Journal of Computer Science & Engineering Technology (IJCSET) ISSN : 2229-3345 Vol. 6 No. 07 Jul 2015.

[10] Abdullah M. Alnajim and Hazim S. AlRawashdeh, "Verifying an Anti-Phishing Model Using Formal Methods" International Journal of Soft Computing Year: 2016, Volume: 11, Issue: 2, Page No.: 76-82.

[11] Kritzinger, E; von Solms, SH. 'E-learning: Incorporating Information Security Governance'. 2006. Accessed Dec 2011. *www.informingscience*.org/proceedings/InSITE2006/ IISITKrit157.pdf.

[12] A. Jalal and Mian Ahmad Zeb. *'Security Enhancement for E-Learning* Portal'. IJCSNS International Journal of Computer Science and Network Security, Vol.8 No.3, 2008.

[13] Najwa Hayaati, Mohd Alwi, Ip-Shing Fan. *'E-Learning and Information Security Management'*. International Journal of Digital Society (IJDS), Volume 1, Issue 2, 2010.