An Experiment Scheduler and Federated Authentication Solution for Remote Laboratory Access

http://dx.doi.org/10.3991/ijoe.v11i3.4554

Ning Wang[1], Xuemin Chen[2]*, Gangbing Song[1] and Hamid Parsaei[3]
[1]University of Houston, Houston, USA;
[2]Texas Southern University, Houston, USA;
[3]Texas A&M University at Qatar, Doha, Qatar

***Abstract*—This paper addresses the needs involved in authentication and authorization when target engineering experiments are exported from the remote experimental server to an end-user experiment scheduler web server (SWS). A solution is discussed based on using a novel unified framework for engineering experiments and integrating MD5 encryption and decryption algorithm-based security management engine for a federated authentication and authorization uniform resource locator (URL). The established remote engineering laboratory at Texas A&M University at Qatar (TAMUQ) demonstrates the efficiency of the proposed SWS and authentication method.**

***Keywords*—remote laboratory; scheduler web server; federated authentication; MD5 encryption; unified framework**

I. INTRODUCTION

The application of remote laboratory technology for enhancing engineering education has attracted much attention in the last decades due to its flexible accessibility and resource sharing [1][2]. The increased use of remote laboratories for online education has made user management and security controlled access issues significant concerns. In order to achieve the goal, easy to manage and control user access to the remote laboratory for using experiments, various solutions have been designed and implemented. In the design and implementation of the scheduler and federated authentication, two critical challenges must be met: 1) how to simply and effectively implement the end-user time management and secure access; and 2) how to achieve high-performance encryption and decryption processes for generating the federated authentication uniform resource locator (URL).

With continuing improvements in computer performance, the technology supporting the browser-server architecture is becoming increasingly more stable and suitable for cross-platform system design. Meanwhile, a large number of new technologies have been developed to support more complex web browser-based Internet applications. Consequently, a number of Remote Laboratory Management Systems (RLMS) have used the browser-server architecture technology. Examples include LiLa (Library of Labs) [3], JT-60U [4], WebLab-Deusto [5], the Networked Control System Laboratory (NCSLab) [6], among others. Although each of these systems are based on a different implementation approach, they have a similar general purpose: to manage and support shared access to a collection of remote laboratories. This shared system access is typically achieved by establishing appropriate access authority for users on the RLMS; after which, they can log in and request access to the experimental equipment. Requests from users can be placed via a user management queue or through making a reservation for access at a predetermined time.

The LiLa project is borne from the alliance of eight European universities and three enterprises. The goal of this project is the composition and dissemination of a European infrastructure for the mutual exchange of experimental setups and simulations, specifically targeted at undergraduate studies in engineering and science. Shibboleth is used as the framework for federated authentication and authorization for the LiLa portal [7]. Shibboleth is a web-based technology, and it implements the HTTP/POST, artifact, and attributes push profiles of Security Assertion Markup Language (SAML), including both Identity Provider and Service Provider components. The JT_60U project, another good example, is the flagship of Japan's magnetic fusion program, produced the federated authentication issue based on ITBL-IS and AEGIS. Any researcher who has successfully passed a screening of the Certificate Authority is allowed to access the remote experiment server through an encrypted channel. Once certified, an off-site researcher can open a web browser by specifying the URL of the remote experimental server. However, almost all of these systems use complex server architecture and specific channels or network ports to implement security control and manage access to the experiment. Additionally, certain numbers of these systems require the support of software plug-ins. To the best of our knowledge, being able to provide a stable and high-performance solution for security controls in remote experiments remains a critical issue.

To solve the challenge of managing, moreover, controlling, end-users access the remote experiments through a simple, stable and high-performance solution, we propose a new solution based on the novel unified framework. This solution includes a scheduler web server (SWS) and a remote experiment access authentication engine. It has been implemented in a hands-on laboratory via an Internet (HLI) project at Texas A&M University at Qatar (TAMUQ). The goal of the HLI project is not only to integrate experiments and laboratories into a browser-server architecture software infrastructure, but also to build a web portal within which experiments and laboratory are provided. This infrastructure includes an online learning system that guides users through experiments. From the technical point of view, the TAMUQ remote laboratory portal is a repository of remote experiments on a central server, which is built on Apache 2.0 web server and Node.js web server based on Centos OS [8]. Providing users an organizational framework for online learning, this remote laboratory portal gives access to these remote experiments within an integrated environment, including a tutoring system for students. Also featured is a local scheduling and user management system for students, teachers, and researchers [9].

To adequately exploit these learning resources, one of the objectives of this project is to provide well-defined access control to remote experiments and remote laboratories. This work addresses the needs of authentication and authorization when reusable experiments are exported from an experiment provider. The project, in essence, becomes an experimental repository that provides access to resources for remote experiments to the user, especially when the experiment is part of a curriculum for students. In this paper, the TAMUQ remote laboratory portal acts as a platform that integrates different modules for laboratory and experiment operating, management, scheduling, authentication and authorization. Also we discuss a possible solution based on the use of the novel unified framework for remote experiments and MD5 encryption and a decryption algorithm-based security management engine for a federated authentication and authorization.

The rest of this paper is organized as follows: Technology used in the scheduler and federated authentication is overviewed in Section II. In Section III, the detailed working processes of the scheduler and federated authentication is presented. In Section IV, the innovative implementation process is presented. Concluding remarks are drawn in Section V.

II. TECHNOLOGY OVERVIEW

A. A Novel Unified Framework

The novel unified framework is based on the Web 2.0 technology, and it includes three parts: client web application, server application, and experimental control application. The client web application is based on Hyper Text Markup Language (HTML), Cascading Style Sheets (CSS), and JQuery/JQuery-Mobile JavaScript libraries and mashup technology is used for user interface implementation. The client web application can be run on most of the current browsers such as IE, Firefox, Chrome, Safari, and others. The server application is based on Web Service technology and is directly built on top of MySQL database, Apache web server engine and Node.js web server engine [10][11]. The server application uses JSON and Socket.IO which is developed based on web socket protocol to implement the real-time communication between the server application and the client-web application [12]. The server application runs on Centos Linux server. The experiment control application is based on the LabVIEW and uses Socket.IO for real-time communication with the server application. The experiment control application runs on experiment control workstation which runs Window 7 OS. Our new unified framework for remote laboratory development is based on three vital technologies, which are the Socket.IO, Node-HTTP-Proxy, and HTTP Live Streaming (HLS) protocol. Node-HTTP-Proxy is used for the novel video transmission approach which is based on HLS protocol for real-time system monitoring [13], and experiment data and control commands transmission and traversing firewall [14]. In addition, a mashup technology for user interface integration is employed in our client applications. A server-based mashup analyzes and reformats the data on a remote server and transmits the data to the user's browser. The architecture of our mashup scheme is divided into three layers:

Presentation / user interaction: this is the user interface of Mashup. Our novel design uses HTML, CSS, JavaScript and Asynchronous JavaScript.

Web Services: the system functionality can be accessed using the API (Application Programming Interface) services. Our novel design uses JSON-RPC, REST, and SOAP.

Data: the data is handled in three ways, i.e., sending, storing and receiving. Our novel design uses JSON and Socket.IO for data transmission.

The system architecture of the new unified framework is shown in Fig. 1.

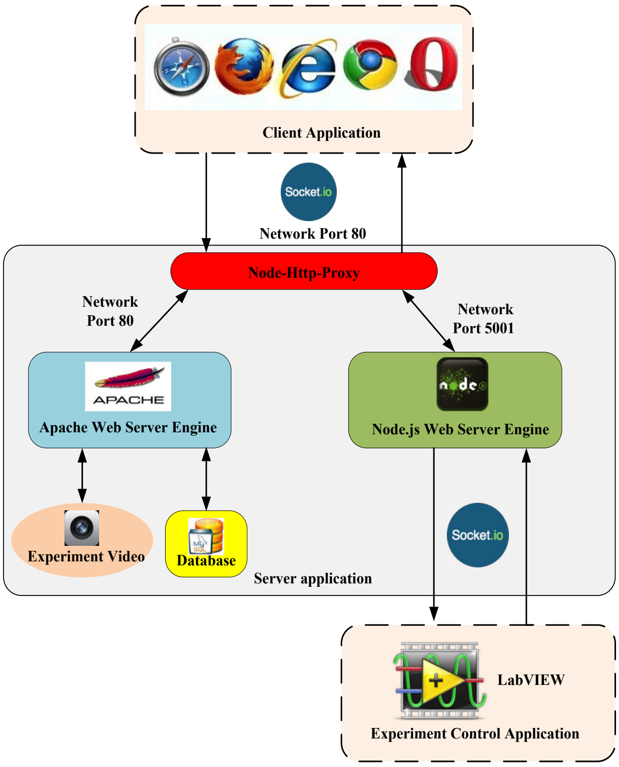


Figure 1. The architecture of the novel unified framework.

With the remote laboratory at TAMUQ, enabled by the novel unified framework, terminal users can conduct or view real-time remote experiments on any device, including portable devices, without firewall issues or need of third party plug-ins for most of Internet browsers except IE which the user needs to install Java runtime engine for real-time vedio display.

B. Uniform Resource Locator

A Uniform Resource Locator is a reference that specifies the location on the Internet and provides a mechanism for retrieving the resource [15]. It has two principal components, protocol identifier, and resource identifier. As shown in Fig. 2, a URL, or web address, is a specific character string that constitutes a reference to a resource. In most of web browsers, the URL of a web page is displayed in an address bar.

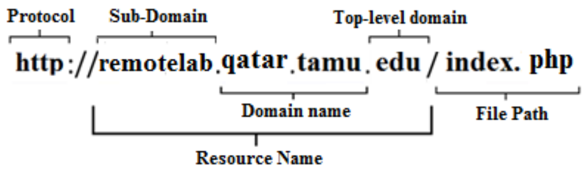


Figure 2. Example of URL.

Although many people use the two terms interchangeably, a URL is a particular type of uniform resource identifier (URI). A URL implies the means to access an indicated resource, which is not applicable for every URI. As shown in the Fig. 3, a URI is either a URL or a uniform resource name (URN), or both.

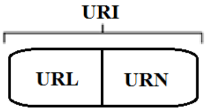


Figure 3. The Relationship of URL, URN, and URI.

The protocol of a URL defines how the resource are obtained. Two common protocols are used for the web resources, and they are HTTP and HTTPS. Each protocol has advantages and disadvantages, for various reasons; as a result, many sites have evolved to permit access through both the HTTP and HTTPS protocols. When a web link contains a protocol specified item, it results in a browser following the link using the specified protocol regardless of the potential desires of the user.

At present, URLs are most commonly used to reference web resources using HTTP protocol, but they are also used for file transmission through the File Transfer Protocol (FTP), database access through the Open Database Connectivity (ODBC) application interface, and many other applications. As users of the Internet are distributed throughout the world and are using a wide variety of languages and alphabets, they expect to be able to create URLs in their vernacular. An internationalized resource identifier (IRI) is a form of URL that includes Unicode characters. All modern browsers support IRIs. The parts of the URL requiring special treatment for different alphabets are the domain name and path. Consequently, it is possible to encrypt the file path within the URL on the web server.

C. MD5 Message-digest Algorithm

The MD5 message-digest algorithm is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, typically expressed in text format as a 32 digit hexadecimal number. MD5 has been utilized in a wide variety of cryptographic applications, and is also commonly used to verify data integrity [16], [17].

MD5 processes a variable-length message into a fixed-length output of 128 bits. The input message is broken up into chunks of 512-bit blocks (sixteen 32-bit words), and the message is padded so that its length is divisible by 512. The padding works as follows: first a single bit, 1, is appended to the end of the message. To bring the length of the message up to 64 bits less than a multiple of 512, it is followed by as many zeros as are required. The remaining bits are filled up with 64 bits representing the length of the original message, modulo 264.

The main MD5 algorithm operates on a 128-bit state, divided into four 32-bit words, denoted A, B, C, and D. These are initialized to certain fixed constants. The main algorithm then uses each 512-bit message block in turn to modify the state. The processing of a message block consists of four similar stages, termed *rounds*; each round is composed of 16 similar operations based on a non-linear function F, modular addition, and left rotation. Fig. 4 illustrates one operation within a round. There are four possible functions F; a different one is used in each round:

$$F(B, C, D) = (B \wedge C) \vee (\neg B \wedge D) \quad (1)$$

$$G(B, C, D) = (B \wedge D) \vee (C \wedge \neg D) \quad (2)$$

$$H(B, C, D) = B \oplus C \oplus D \quad (3)$$

$$I(B, C, D) = C \oplus (B \vee \neg D) \quad (4)$$

$\oplus$, $\wedge$, $\vee$, $\neg$ denote the XOR, AND, OR and NOT operations respectively.

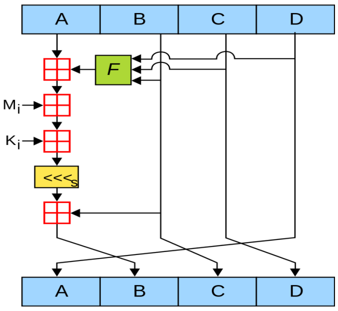


Figure 4. MD5 operation.

MD5 message-digest algorithm has been widely used in the software development to provide some assurance that a transferred file has arrived intact and the federated authentication control. In our project, the remote laboratory center server provides a pre-computed MD5 (well known as MD5 sum) checksum for the remote experiment web page URL, so that the user side web application can compare the checksum merged in URL to open the remote experiment control web page.

D. Scheduling Server Based on Global Server Architecture

To integrate the current three remote laboratories (Rlabs), i.e., Rlab at TAMUQ, RLab at University of Houston (UH) and RLab at Texas Southern University (TSU), into one global level remote laboratory, global server architecture is designed for addressing this issue. In this global level remote laboratory, a global server has been set up and worked as a center proxy to connect the other two lab center servers. Through the global server, the terminal users can use all remote experiments in these three universities. In the global remote laboratory, there are also a unified scheduling server and user management system. Fig. 5 shows the global remote laboratory architecture.

The scheduling server is the important component normally used to manage the scheduling processes of the remote laboratory access and operation, including tracking the state of operations and assigning them to users based on either queued access or time-based reservations. Meanwhile, it is also responsible for managing the running sessions according to the allocated times as well as controlling all events and activities.

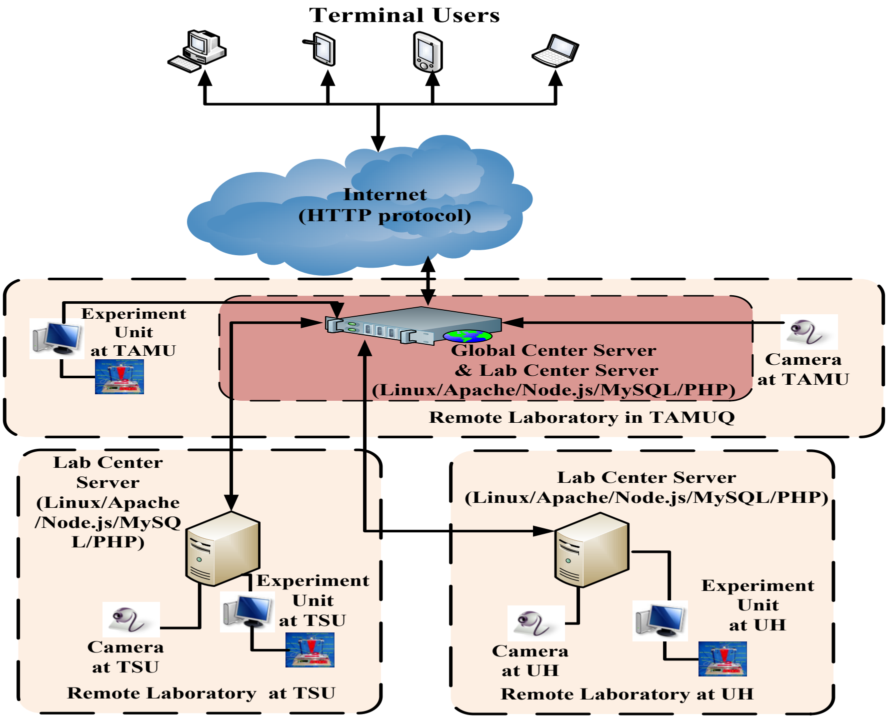


Figure 5. The global remote laboratory architecture.

When an end-user is requesting to use an experiment, it is the scheduling server that ultimately decides whether to queue the user, as well as if or when they are assigned to an experiment operation and an experiment time sheet. In the novel unified framework, the scheduling server already has a programmatic interface to provide these functions through an API. Since the confirmed process of authentication is finished upon the Web Interface (normally the web page) in the novel unified framework, the scheduling server is only used to make an authorization decision for end-users.

Depending on the independent demands made by different remote laboratory systems, the scheduling server is made the specific design relying on the user's individual behavior in accessing and operation.

III. SCHEDULER AND AUTHENTICATION

To avoid copying the remote experiment control page URL into a new browser to open a web page, The HLI project uses the authentication and authorization URL to control user access. One possible solution is to use the MD5 encryption and decryption algorithm to design and develop an authentication engine to implement this objective. As shown in Figure 6, the MD5 encryption and decryption algorithm-based security management engine includes two parts: the MD5 encrypted engine in server and the MD5 decryption module in client. This section discusses the authentication and authorization mechanisms used in HLI project and our realization solution to overcome this problem that is to avoid copying the remote experiment control page URL into a new browser to open a experiment control web page. To control all remote experiments and terminal users, a global web scheduler server and several local web scheduler servers also are built up based on the LANMP (Linux/Apache/Node.js/MySQL/PHP) web server. Each local server can be configured to share the resources locally, and can share the resources globally through the global server also.

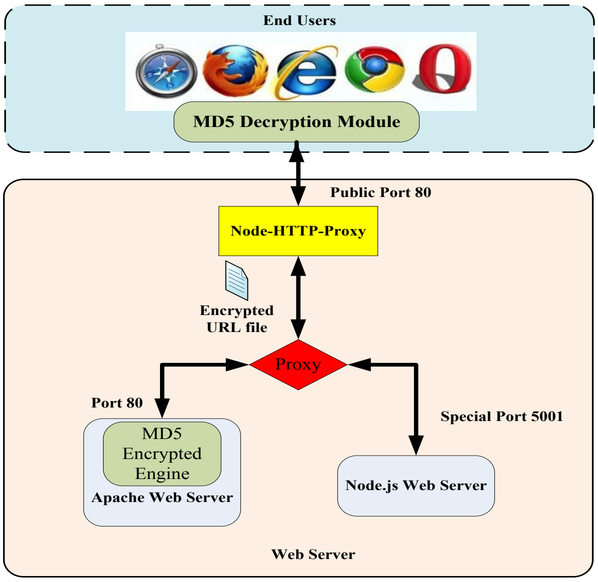


Figure 6. The HLI authentication architecture.

A. *The HLI Authentication and Authorization Architecture*

MD5 encryption and decryption algorithm-based security management engine are designed and developed in the HLI project at TAMUQ to control access by registered users to the portal as well as to control access to experiments and remote laboratories. Concerning authentication and access control, the general portal software architecture, based on the novel unified framework, is also shown in Fig. 6.

The general process is as follows:

A user goes to the home page of the TAMUQ remote laboratory portal and, from this page, the user can log into the user management system.

To access the experimental control portal, the user must log into the laboratory scheduler to book the remote experiments based on available time slots.

When the user booked experiment is ready, the authentication process is enabled. As a result of the authentication process, the user receives a remote experiment control page URL that is encrypted in remote laboratory center server, and the user's web browser will checksum the MD5 encryption URL through comparing a key that is saved in a database on the server. These attributes are used to allow or deny user access to protected resources.

If the URL is authenticated and the user has the required permissions, users will go to the remote experiment control page where they can conduct the experiment in real time; meanwhile, they also can simultaneously see the experimental video.

B. Server Side MD5 Encrypted Engine

To generate the authentication and authorization URL for the remote experiment control page, an encrypted engine with MD5 encryption and a decryption algorithm is developed.

As the MD5 message-digest algorithm defined, it can produce a 128-bit (16-byte) hash value and generate an encrypted key with a fixed-length output of 128 bits. The URL generation engine is implemented based on this algorithm using PHP language on the server-side. For the encrypted URL generation, the users' requested time is used for the input hash value. The PHP system time function, *"gettime()",* is used to get the system clock, and then put this value to the engine to generate an encrypted key. The key is merged to the original sample URL and the server feeds back the new generated encrypted URL to the user.

Because the system clock of the user's requested time is the unique value, the MD5 encrypted key generated by this unique input value is also the unique key. This solution is not only difficult to crack, it is also easy to control and use in the client side decryption process.

C. Client Side MD5 Decryption Module

As the authentication and authorization URL needs to be sent to the client side to decrypt, the MD5 decryption module also must be designed and developed for the client side. To achieve this goal, the PHP language also must be used to implement the MD5 decryption algorithm. In most cases, the MD5 message-digest algorithm is paired design. To open the remote experimental control web page, the MD5 decryption module will parse out the available URL, which can be identified by web browsers.

The end-users who seek access to the available remote experiment control web page also require permissions. At first, these users must log into their accounts which are managed by the user management system included in the remote laboratory system at TAMUQ. After the end users receive the available URL, they can open it within the allowed time slot, currently set to 10 seconds. However, based on differing system requirements, the effective time slot has a flexible configuration. The Fig. 7 shows detail working process of the MD5 encrypted engine.

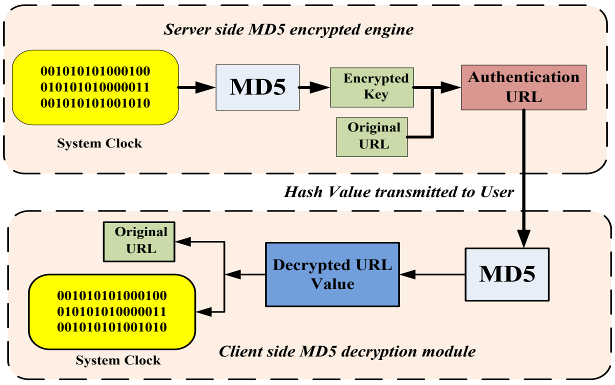


Figure 7. Working process of the MD5 encrypted engine.

D. Scheduler Web Server

The scalable Scheduler Web Server will be redesigned by using REST-driven web services. First and foremost, the client web interface will be written in JavaScript and HTML to allow for dynamic content generation, and it will be hosted on the SWS. Afterward, the SWS opens a communications port to the experiment to proxy the exchanged user data following the REST methodology. The proxy service allows the client to exchange data with the remote experiment without introducing data verification or authentication from both client and experiment sides. The server will verify the data follows the developer configuration file and the user is authorized to interact with the experiment. In addition to the proxy service, the SWS will provide user, experiment, and data management.

The SWS manages all the authentications for the users and experiments as shown in Fig. 8. After the Internet user finishes loading the experiment interface, the JavaScript software running on the client's computer initiates a communication channel with the server and sends back the session ID and IP address of the client. If the server recognizes the session ID and client IP associated with the user login name and password as a person who has current access to the experiment requested, then the server initiates a new communication channel with the experiment. At this point, the experiment and the client communicate through the server to exchange data. Since the data exchanged with the client pass through SWS, the server will always be in charge of checking and verifying that the user is allowed to run the experiment.

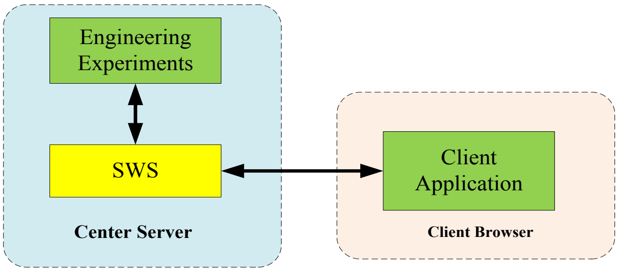


Figure 8. Working process of the SWS.

IV. SOLUTION IMPLEMENTATION

A. *Technical Characteristics of the Federated Authentication URL Encrypted Engine*

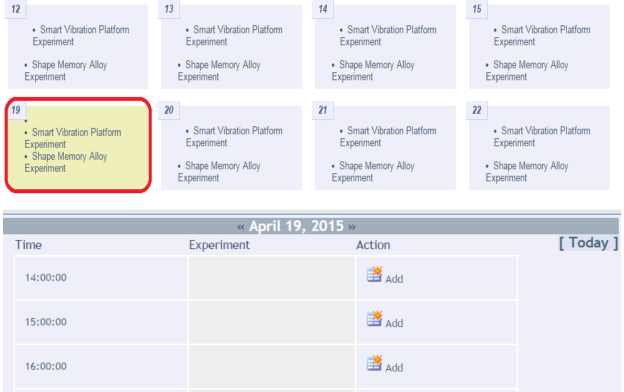
The architecture of the federated authentication URL encrypted engine, which is described in the section III, has been used for the new Smart Vibration Platform (SVP) remote experiment and the new Smart Memory Alloy (SMA) remote experiment. These two remote experiments has been developed based on the novel unified framework. As illustrated in the section II, a scheduling server based global server architecture is designed. With this architecture, a global web scheduler server is implemented in the HLI project at TAMUQ. Meanwhile, for user access security control, the federated authentication URL encrypted engine is designed and implemented as well. These two modules are a significant improvement for the novel unified framework.

To implement the global server architecture, there are three local scheduler web server which are built up based on the LANMP server architecture at TAMUQ, UH and TSU. At these three universities, the local scheduler web server manages the different remote experiments developed in their own laboratory. The global scheduler server is used to integrate all of these remote experiments together. It means that the end users can use all of the remote experiments settled at three different universities only through one user interface on the global scheduler server. The PHP language is used to implement the global scheduler web server and the global scheduler web page is integrated with mashup technology.

To implement the remote experiment control web page URL management, a new URL management module is developed with PHP and MySQL database-driven module. Meanwhile, two data table are designed and developed based on MySQL database. To develop the authentication and authorization URL encrypted engine on the server, the encryption and decryption module is developed based on the Md5 algorithm with the PHP language. The engine is run on the LANMP architecture web server. The user access date and time are used as the input to the engine, and then combined the key to the experiment control page URL to generate the new authentication and authorization URL.

B. *A New Scheduler Wed Server and User Management System*

To manage the remote experiments, a new scalable experiment scheduler system and a new user management system were implemented. The Representational state transfer technology was used to design the architecture of SWS. XML HTTP Request (XHR) is used for the communications function implementation between SWS and the client side. XHR is a set of Application Programming Interface (API) functions, available in most web browsers, capable of initiating an HTML request from the JavaScript running on the Client computer, and updating the Document Object Model (DOM) in real time without reloading the entire web page. With the LANMP server architecture, the web scheduler server was built up. The PHP language was used to implement the server side systems. Meanwhile the MySQL 5.5 database system was used to manage the experimental data, user information, and other system information. For the client side user interface implementation, the HTML, CSS, and JQuery/JQuery-Mobile JavaScript libraries were used. Fig. 9 shows the screenshots of the scheduler, the experiment conflict control, and the experiment management page.


12
- Smart Vibration Platform Experiment
- Shape Memory Alloy Experiment

13
- Smart Vibration Platform Experiment
- Shape Memory Alloy Experiment

14
- Smart Vibration Platform Experiment
- Shape Memory Alloy Experiment

15
- Smart Vibration Platform Experiment
- Shape Memory Alloy Experiment

19
- Smart Vibration Platform Experiment
- Shape Memory Alloy Experiment

20
- Smart Vibration Platform Experiment
- Shape Memory Alloy Experiment

21
- Smart Vibration Platform Experiment
- Shape Memory Alloy Experiment

22
- Smart Vibration Platform Experiment
- Shape Memory Alloy Experiment

« April 19, 2015 »

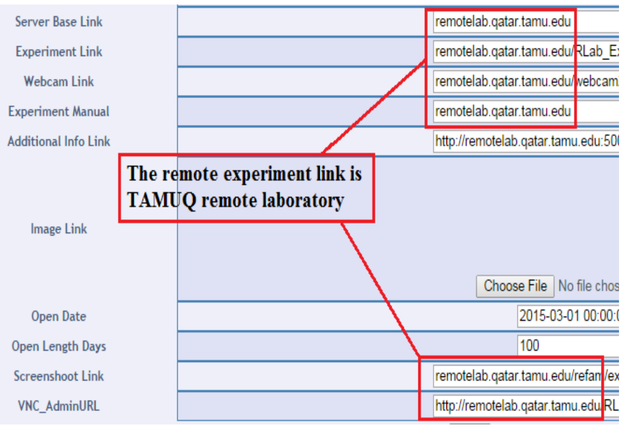
[Today]

Time	Experiment	Action
14:00:00		Add
15:00:00		Add
16:00:00		Add

Figure 9. Scheduler and experiment conflict control page screenshots.

C. *Sample Paradigm of the Federated Authentication Solution*

The federated authentication allows students of one college to use their authentication credentials to access remote experiments that are set up on another college campus. In Fig. 5, the global remote laboratory architecture is designed, and a pilot case has been implemented at TSU.



Server Base Link	remotelab.qatar.tamu.edu
Experiment Link	remotelab.qatar.tamu.edu/RLab_E
Webcam Link	remotelab.qatar.tamu.edu/webcam
Experiment Manual	remotelab.qatar.tamu.edu
Additional Info Link	http://remotelab.qatar.tamu.edu:50
Image Link	Choose File No file chos
Open Date	2015-03-01 00:00:
Open Length Days	100
Screenshoot Link	remotelab.qatar.tamu.edu/refan/ex
VNC_AdminURL	http://remotelab.qatar.tamu.edu/RL

Figure 10. Remote experiment configuration in VR-Lab at TSU.

Fig. 10 shows the configuration interface of the scheduler at TSU, which is implemented under the novel unified framework to support all remote experiments. This scheduler configuration web page can be accessed using the administrator authority in all common web browsers running on any terminal device without the installation of plug-ins. As an example, the SVP remote experiment at TAMUQ is configured in the scheduler server at TSU. After the configuration, the SVP experiment at TAMUQ web page can be accessed using the authentication URL through the scheduler and user management system at TSU.

The global remote laboratory architecture based on the novel unified framework is capable of taking full advantage of the previous remote laboratory system. With the federated authentication URL solution, the improved remote experiment provides better security access control for a wider range of terminal equipment.

V. CONCLUSIONS

This paper discussed a solution for the experiment scheduler and federated authentication needs of experiments used to control and access remote experiments in the context of the Hands-on Laboratory via Internet project at TAMUQ. The scheduler and federated authentication solution was designed and implemented. The security control and management of experiment access interface issues were well-solved by taking full advantage of the functionalities offered by the MD5 encryption and a decryption algorithm-based security management engine. Consequently, the scheduler and federated authentication solution for web applications give the unified framework much needed improvement.

REFERENCES

[1] L. Gomes and S. Bosgoyan, *"Current trends in remote laboratories"*, IEEE. Trans. on Industrial Electronics, Vol 56, NO 12, pp. 4744-4756, December 2009, ISSN: 0278-0046.

[2] J. Rodriguez-Andina, L. Gomes and S. Bogosyan, *"Current trends in industrial electronics education"*, IEEE. Trans. on Industrial Electronics, Vol 57, NO 10, pp. 3242-3244, October 2010, ISSN: 0278-0046.

[3] T. Richter, D. Boehringer , S. Jeschke, *"LiLa: A European Project on Networked Experiments"* Automation, Communication and Cybernetics In Science and Engineering , 2009, Part 2, 307-317, http://dx.doi.org/10.1007/978-3-642-16208-4_27

[4] T. Ozeki, Y. Suzuki, T. Totsuka, K. Iba, S. Sakata, N. Miyato, A. Isayama et al. "Development and Demonstration of Remote Experiment System with High Security in JT-60U." In 22th IAEA Fusion Energy Conf., Geneva. 2008.

[5] P. Orduna, J. Irurzun, L. Rodriguez-Gil, J. Garcia-Zubia, F. Gazzola, and D. Lopez-de-Ipina, "Adding new features to new and existing remote experiments through their integration in WebLab-Deusto," Int. J. Online Eng., vol. 7, no. S2, pp. 33–39, 2011. http://dx.doi.org/10.3991/ijoe.v7is2.1774

[6] W. S. Hu , G. P. Liu & H. Zhou *"NCSLab: A web-based global-scale control laboratory with rich interactive features"*. IEEE Transactions on Industrial Electronics, ISSN: 0278-0046, 57(10), 3253-3265. 2010

[7] L. Bellido, V. Villagrá, and V. Mateos. "Federated authentication and authorization for reusable learning objects." In Education Engineering (EDUCON), 2010 IEEE, pp. 1071-1074. IEEE, 2010. http://dx.doi.org/10.1109/educon.2010.5492459

[8] X. Chen, D. Osakue, N. Wang, H. Parsaei, G. Song. "*Development of a remote experiment under a unified remote laboratory framework*," QScience Proceedings (World Congress on Engineering Education 2013) 2014:7, http://dx.doi.org/10.5339/qproc.2014.wcee2013.7

[9] N. Wang, X. Chen, G. Song and H. Parsaei, *"Remote Experiment Development Using an Improved Unified Framework"*, in Proceedings Of AACE E-Learn 2014-World Conference on E-Learning, New Orleans, LA, United States, October 27-30, 2014.

[10] T.Hughes-Croucher and M. Wilson, *"Up and Running with Node.js (First ed.), "* O'Reilly Media, 2012, p. 204, ISBN 978-1-4493-9858-3.

[11] D. Herron *"Node Web Development, Second Edition"*, Packt Publishing, ISBN 184951514X, Jul 19, 2013.

[12] R. Rai *"Socket. IO Real-time Web Application Development"*, O'Reilly Media , ISBN 178-2-1607-87, February, 2013.

[13] N. Wang, X. Chen, G. Song and H. *Parsaei "A Novel Real-time Video Transmission Approach for Remote Laboratory Development."*International Journal of Online Engineering (iJOE), vol. 11, no. 1 (2015): pp-4.

[14] N. Wang, X. Chen, G. Song and H. Parsaei *"Using Node-HTTP-Proxy for Remote Experiment Data Transmission Traversing Firewall."*International Journal of Online Engineering (iJOE), vol 11, no. 2 (2015): pp-60.

[15] M., Larry, T. Berners-Lee, and R. T. Fielding. "Uniform resource identifier (URI): Generic syntax." (2005).

[16] J. Black, M. Cochran, and T. Highland . *"A study of the MD5 attacks: Insights and improvements. "* In Fast Software Encryption . Springer Berlin Heidelberg. 2006, January. pp. 262-277.

[17] M. Ciampa, *"Comptia security+ 2008 in depth."* Cengage Learning. 2009.

AUTHORS

Ning Wang is with Department of Electrical and Computer Engineering, University of Houston, Houston, TX, USA (nwang@uh.edu).

Xuemin Chen is with Department of Engineering, Texas Southern University, Houston, TX, USA (e-mail: chenxm@tsu.edu). He is the corresponding author of this paper.

Gangbing Song is with Department of Mechanical Engineering, University of Houston, Houston, TX, USA (e-mail: gsong@uh.edu).

Hamid Parsaei is with Department of Mechanical Engineering, Texas A&M University at Qatar, Doha, Qatar (e-mail: hamid.parsaei@qatar.tamu.edu).

This article is an extended and modified version of a paper presented at the International Conference on Remote Engineering & Virtual Instrumentation (REV2015), held in Bangkok, Thailand, 25 - 28 February 2015. This work is based upon work supported by the Qatar National Research Fund under Grant No. NPRP 4-892-2-335. Submitted 22 March 2015. Published as resubmitted by the authors 04 May 2015.