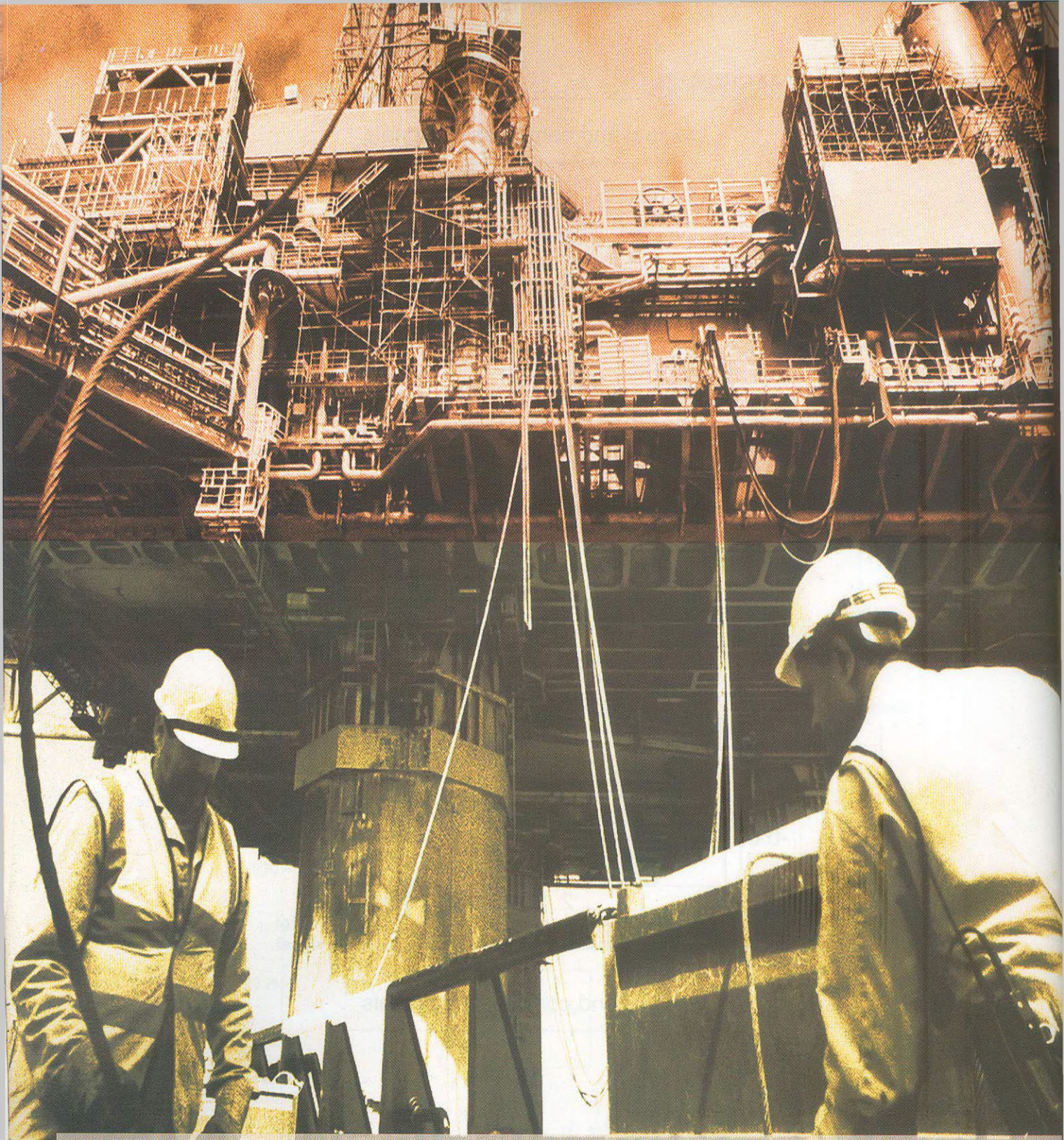
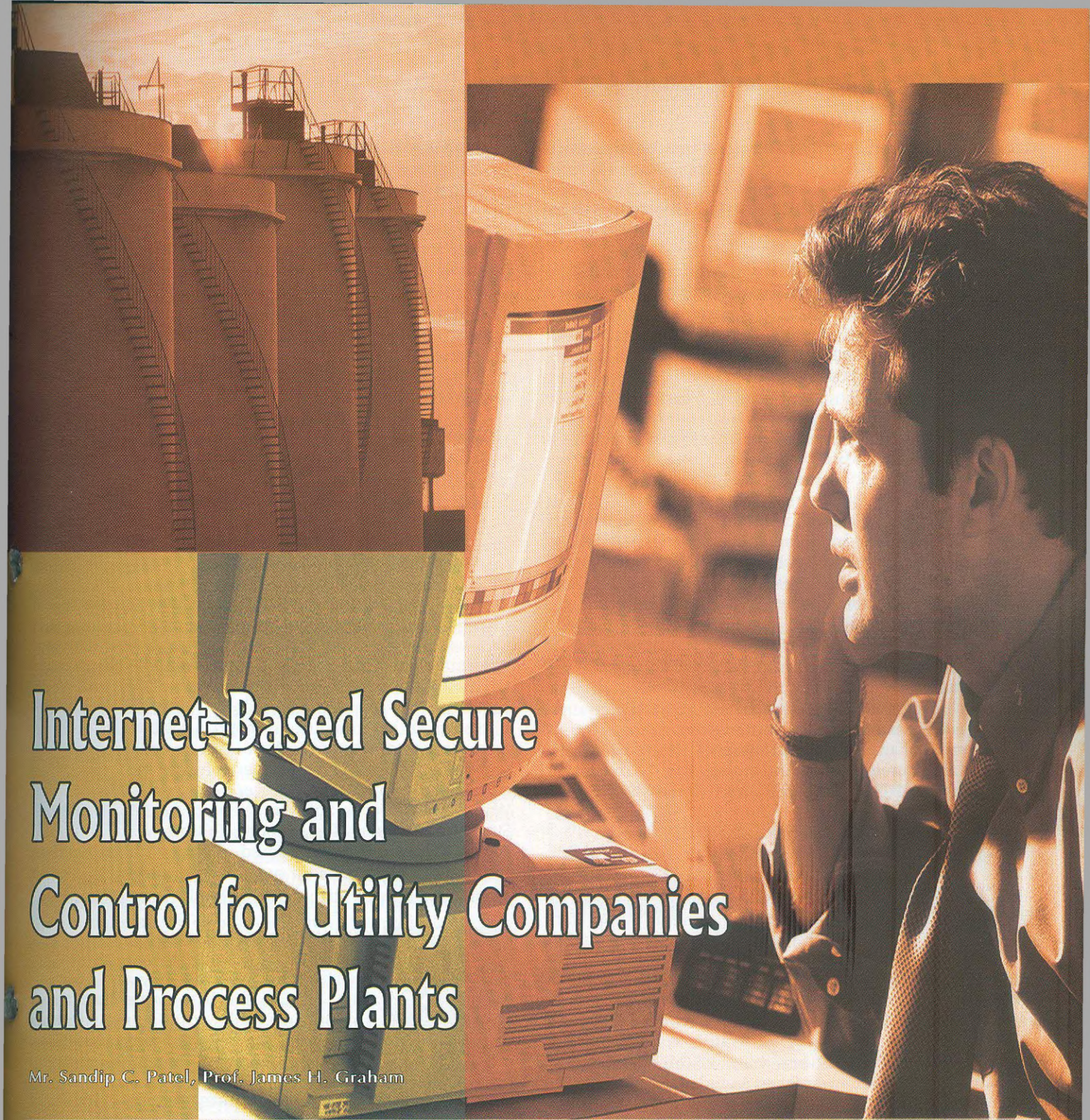


ABSTRACT

Supervisory Control and Data Acquisition (SCADA) networks are used by many of the utility companies that form the critical infrastructure in many countries to control field devices from a central station. In recent years, SCADA networks have also enjoyed an increase in popularity within process control industries for monitoring and control. The use of the Internet can facilitate the remote monitoring of the SCADA systems but with the sharp increase in actual and potential threats against SCADA has made it dangerous to use the unsecured communication protocols. This paper presents a secure architecture to access SCADA-controlled devices remotely using the Internet. Using this architecture, SCADA engineers can get the current process information and send the control data to a remote plant through their web browser securely. This Internet-based approach has been implemented and tested for remote monitoring and control of a chemical-process test-bed.



Internet-Based Secure Monitoring and Control for Utility Companies and Process Plants

Mr. Sandip C. Patel, Prof. James H. Graham

INTRODUCTION

The Supervisory Control and Data Acquisition (SCADA) networks are used to perform critical functions for industrial measurement and control systems and are commonly used by utility and infrastructure companies for monitoring and control of the generation and transition of products such as electric power, oil, gas, and water. They are also commonly used in chemical production and processing plant, other manufacturing plants, and for control of railroads and mass transit. SCADA networks enable industrial measurement and control for these utilities from a central station by enabling remote monitoring and control of multiple field devices, for example, valves, pumps, traffic signals, and electric circuit breakers. SCADA-based system can also provide alarms and some degree of automation by making control decisions using the master computer. Today, SCADA systems have gained much popularity with advancements in network technology (such as TCP/IP which is used for the Internet technology), and wireless technology and with an ever-increasing amount of equipment available for implementing these networks. Additionally, the increased need to monitor the industrial production facilities, such

as at a chemical plant, remotely and as automatically as much possible has fueled SCADA popularity. Global revenues from sales of control systems are predicted to grow from $10.3 billion in 2000 to $12.4 billion in 2006 to $13.9 billion in 2009 by a market research firm Datamonitor (Geer, 2006).

Modern SCADA networks are integrated with corporate networks and the Internet. Field data is transmitted over a wide range of communication lines. Communication between such integrated system elements often uses Ethernet and Internet protocols and other technologies. Network enabled devices, routers, switches, and Window-based operating systems are now quite common in SCADA systems, bringing with them the vulnerabilities that are experienced in desktop computers and corporate networks. A terrorist attack on a SCADA network could cause extreme destructive consequences to public health and safety.

The related work on Internet-based process control is either (1) commercial, mostly implemented as a small extension to the proprietary products, such as JPC (Java for Process control) by NetModule Inc. (JPC), LabVIEW from National Instrument Corporation (National Instrument), iWebServer (iFix brochure) as part of iFIX by GE Fanuc, (2) academic (Tantalean, 2003) but do not use SCADA protocol for its communications and/or have no security features or, (3) aimed towards remote monitoring capacity only. In this paper, we present an Internet-based secure SCADA system that makes it possible to get the current process values from a remote terminal unit (RTU) and display them on a web browser, and securely send control messages to the outstation from the same web browser.

YSTEM REQUIREMENTS

The important function of a SCADA system include the display of information such as reservoir level, feeder current, pump state and issue the controls by the central station or the master terminal unit (MTU) in response to information gathered by the outstations or the RTU. Secondly, it must display the vital information clearly and without error. In many systems real-time display is not a primary function (especially in Water Systems). The requirements of the Internet-based secure SCADA monitoring and control system that we designed included the following:

- User interface should display important current operating parameter values (such as set points, controller tuning and it should be user-friendly, visual, and clear.
- The displayed parameters should be continuously updated with the current information.
- The time interval for polling (asking field devices if they have any data to transmit) for the system should be configurable.
- Users should be able to enter the control data in one screen rather than taking the user through multiple web pages.
- Users should be able to monitor and control in the same window.
- The system should be implemented by using a standard protocol.
- The system should provide end-to-end security by means such as secure SCADA protocol.
- The system should provide secure and administrable front-end.
- The standard technology should be used in the implementation.

The Java programming language was selected since Java provides an operating-system independent platform and does not use proprietary technology such as ActiveX. Java uses open standards such as TCP/IP, works well with client/server technology, provides good networking utilities, and works well with the Internet technology. Additionally, the Java Industrial Automation Extension API is predicted to open the door for development of SCADA software application in Java (Palu, 2002). For the web-server software, Apache, Tomcat and Internet Information Server were considered. Tomcat Web Server (Apache Software) Version 4.1.29 was selected. One of the reasons for selecting Tomcat was that Tomcat supports Java and its related tools, such as servlet. Additionally, Tomcat is an open (non-proprietary) product that is compatible with the latest technologies and standards.

ECURE-COMMUNICATION COMPONENTS

The secure architecture has three components, as shown in Figure 1, which are connected by two-way communication. The user component, the communication support component and the process-control component. In this architecture the current information from industrial process controls is displayed on the user's web browser, and

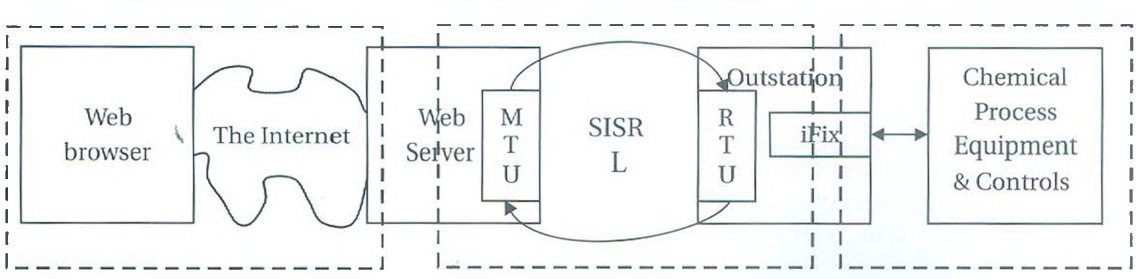
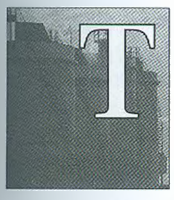


Fig. 1 : Secure Internet-Based SCADA Architectural Components

the set point sent by the user through a web browser is sent to the control system. The following subsections describe these components and the security enhancements implemented within them.

HE USER COMPONENT

The user component includes the interface to human operators who are in charge of monitoring or controlling the SCADA process. By entering the commands such as set points and monitoring the real-time process parameter values on their web-browser screen, the proposed system users, such as operators and engineers, interact with the user component to control and monitor the SCADA process. Providing this human-machine interface is one of the functions of the user components. Since the proposed system is available via the Internet, it can be practically available from any geographical location. The user component has been enhanced with a secured front-end which implements the security mechanism via user authorization. Additionally, the operators/engineers can be given different privileges depending on what they are authorized to do. The privileges such as controlling the process (for example, sending a set point value to the process) can be limited to only certain personnel.



HE COMMUNICATION - SUPPORT COMPONENT

The communication-support component of the proposed system transfers data between the web server and the outstation using Distributed Network Protocol (DNP3) SCADA protocols (DNP3 Organization, and Graham, 2005 in DIASTR) over TCP/IP. It sends the control data entered by users to outstation and sends real-time parameter values from the outstation to the web server to have them display on the user human-machine-interface. These functionalities have been implemented by SCADA Simulator developed at the Intelligent Systems Laboratory (SISRL) at the University of Louisville. SISRL assists in message exchanges simulating DNP3 SCADA communication protocol. The SISRL software simulates communications between a SCADA master and a SCADA outstation by its two software modules: RTU and MTU using a client-server architecture. In the proposed system architecture, the web server is treated as the master.

A web server performs web-hosting tasks and supports graphical user interfaces. The proposed system makes use of Java servlets to update web contents dynamically. The web server also acts as a database server in the proposed system architecture. It stores the data sent from the outstation and those sent by the user. The web-server software include (1) a database system to read and write control and display data in a file on the computer, (2) a server, Apache Tomcat, to enable the computer as a web server, (3) servlet programs written to send/receive information from/to the web browser, (4) the front end secure authentication programs and MySQL open source database to run with them, and (5) the MTU component of SISRL to handle message communications.

Three security mechanisms for SCADA, proposed by Graham et al. (Graham, 2005 in Proc. of WMSCI), were implemented. These security enhancements include Secure Sockets Layer/Transport Layer Security (SSL/TLS), authentication via digital signatures, and authentication via challenge-response. The SSL/TLS solution (Graham, 2005 in DIASTR) to the protocol security provides a fast, standard, and economical solution. The test-bed implementation of SSL/TLS uses Java Secure Socket Extension. SCADA vendors can choose to use the public domain toolkits such as OpenSSL, a commercial-grade, full-featured SSL/TLS toolkit, to implement SSL/TLS. The digital-signature enhancement provides authenticity of the origin and the content of the message. The implementation of this solution calculates a hash digest on the message that is being sent, encrypts the digest with MTU's private key and sends this encrypted digest to an RTU along with the message. The RTU uses MTU's public key to decrypt the value. The authenticity of origin is proven if the RTU could successfully decrypt the value. RTU calculates the hash value and compares it with the digest it received. If the hash values match, the authenticity of the message-content is proven. The challenge-response algorithm periodically verifies the identity of the devices (master or an outstation) by using the challenge-response cryptography to protect against the man-in-the-middle attack. Either of the devices can initiate the challenge. The authenticator sends a random "challenge" message to the other device to which the device must respond with a value calculated using a hash function. Only the valid devices can calculate a correct hash since the hash stream contains a "secret", known only valid devices.

HE PROCESS COMPONENT

The process component varies with the application. The following section discusses a process component implemented at the SCADA test-bed for the proposed system at the University of Louisville Intelligent Systems and Process Control Laboratories.

CASE STUDY

Figure 2 depicts the details of the chemical process components used in the SCADA test-bed. The real-time information from the water-level control system is sent to the user's web browser and the set points sent by the user are inputs to the water-level control system. The process component (1) reacts to the user control input, and (2) acquires and sends the real-time parameter values to the users.

The water-level control system is a process controlled automatically by SCADA software. iFix32® Version 7.0 from Intellution Inc. (available through GE Fanuc) is used on a PC with Windows 2000 acting as a remote terminal unit. The hardware for the water-level control system include a terminator panel OPTO 22 ®, and the process equipment. The terminator panel links the computer's serial port with the process' measurement and two actuator devices: (1) Channel 5: 4 to 20 mA signals used for controlling analog input (Analog to Digital), and (2) Channel 4: 0 to 10 volt signals used for

controlling analog output (Digital to Analog). The process equipment include water-lines, an electrical pump, reservoir, "tan made up of three vertically connected glass flasks, and controls associated with the lines such as sensors, a manually controlle outflow valve and a SCADA controlled inflow valve.

The control objective of the test-bed process is to maintain the specified water level in a tank. Water from a reservoir is passe

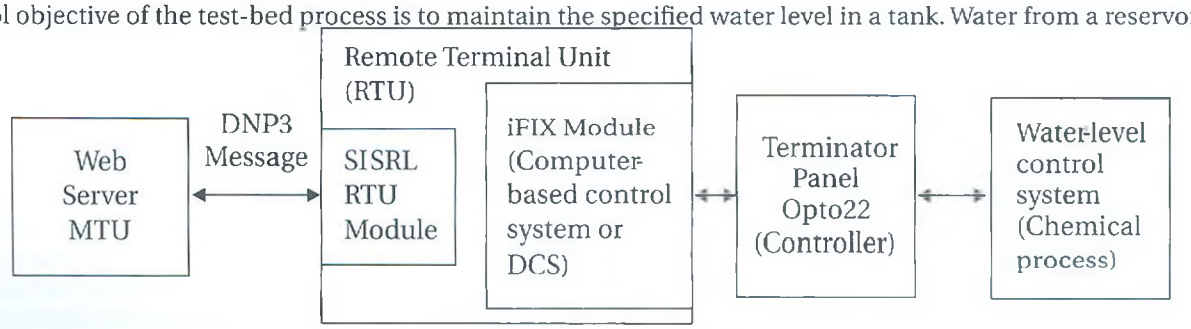


Fig. 2: Case Study Component Architecture

through a valve, which is automatically regulated by the SCADA controller, to this tank. The outflow from the tank to the reserv is manually controlled by a valve, which needs to be set only once (when the SCADA operation is started). In order to maintair constant set/desired level, the SCADA have to make calculations and then send control commands accordingly to the inflow val to open or close it to increase or decrease the water-level in the tank. That is, a human operator controls outflow manually keeping outflow valve slight open and dripping water into the reservoir while SCADA controls the inflow valve automatically iFix32®. Details on the process operation are published by Patel et al. (Patel, 2005) and are available on-line.

Using the user component discussed in the above section on the user component, a user enters the set point value and hits t "Update" button located in the left panel of the web browser. Java servlets send this value to the water-level control system RT The right panel shows the real-time parameter values including the valve position in terms of percent open, the water level in t tank, and the set point value. A PC running Windows XP operating system with 1.0 GHz Intel® Pentium® IV processor having 2 MB RAM is used as an MTU and a 350 MHz PC running the Windows 2000 operating system and having 400 MB RAM is used as t RTU.

ESTING AND RESULTS

The water-level control set-up and the outstation are located in the Process Control Laboratory in the Chemical Engineering building at the University of Louisville. The web server is located in the Intelligent Systems Research Laboratory in the J.B. Speed School building. The control and display functionalities of the proposed system have been tested thoroughly. First, as a unit test, SISRL was tested successfully by passing DNP3 messages between the web server (treated as the master) and the outstation (treated as an RTU). After testing various elements of the front-end, the user component was also successfully tested by logging into the system from an off-campus laptop connected to the Internet. Finally, fu functional Internet-based system implementation was test using the University of Louisville Intranet and the Intern from web browsers using different computes.

Following the completion of the functional testing describ above, performance testing was executed to study the tir spent in the message communications. The study includ time delays for the communication between (1) a web brows and the MTU, and (2) MTU and RTU with and without t overheads caused by the three SISRL security enhancemer described in the section on the communication-suppo component. For each of the performance measuremen twenty independent trials were run to get a good coverage different network and operating-system conditions. The del

	Message-communication time	Reading Range	Standard Deviation
DNP3	325	(286, 357)	13
DNP3 + SSL/TLS	373	(300, 403)	34
DNP3 + Digital Signature	2146	(2039, 2250)	60
Challenge Response Message	446	(369, 588)	49

Table 1: Summary of Performance Results (Roundtrip times in Milliseconds)

between a web browser and the MTU is the latency between initiating a request and displaying the real-time control parameters received from the MTU. This delay, which includes the Internet-communication delays, was measure to be 31 milliseconds with readings varying from 12 to 59 milliseconds. The overall round-trip communication times between the MTU and the RTU for DNP3 without any security enhancement, and with the three enhancements are summarized in Table 1.

MPLICATIONS OF THE TEST RESULTS

The above performance results are acceptable when it is noted that the proposed system is designed as monitoring and supervisory control protocol and not as a protocol for real-time control. Additionally, SCADA companies can decide to implement one or more security enhancements evaluated in this research. Using this research, they can make informed decisions based on what performance cost they are willing to pay for the additional security they buy. Historically, SCADA communications required only reliability and operability using poor communication (e.g., 1200-baud lines) without demanding speed as a criterion. Most SCADA devices are normally polled at every few seconds (Su, 2005) or even every few hours. Considering these long scan-cycle times, the cost of the security enhancements over a period of time would be quite small. For example, if polling is done every 10 seconds, an addition of, say, 500 milliseconds would be only ½ a percent of the 10-second polling-duration. To most companies needing higher security, this overhead would not be prohibitive.

SCADA companies very concerned about performance can choose to implement only the SSL/TLS solution which added only 48 milliseconds per message in these experiments. Comparison of the security enhancements showed that encryption was very expensive compared to the hash both in terms of messaging and computation. The digital-signature enhancement used both hashing and encryption which added 1821 milliseconds per message. This cost would probably be acceptable to the SCADA companies requiring high security but not very concerned about performance. For this enhancement, the companies could choose to add the digital signature only the control messages instead of all the messages and thus reducing the performance degradation. Additionally, encryption hardware can be used to improve the performance. The challenge-response security enhancement provides flexibility in terms of frequency of the challenge messages. The companies could decide how often they would like to send a challenge to verify device identity and thus control their performance cost.

ONCLUSION

The SCADA systems, which are distributed network of sensors and switches that can be controlled remotely to maintain large-scale infrastructure such as pipes and transmission lines, are widely used for measurement and control of industrial systems from a central station. This paper proposes architecture for secure SCADA communications over the Internet. The objective of the proposed architecture is to enhance, and not replace, the computer-based process control systems by providing an extra facility to monitor and control such processes at multiple locations. This Internet-based SCADA system makes it possible to get the latest values from an outstation and display them on a web browser, and to send the control messages to the outstation from a web browser, using a web-based client/server architecture that incorporates DNP3 SCADA communication protocol. The proposed system has been implemented and tested at the University of Louisville test-bed enabling the monitoring and control of a SCADA system that runs a simple chemical process. The testing of this system met all requirements showing that it has a great potential for practical applications to many industrial SCADA systems for monitoring and display of the industrial parameters.

REFERENCES

1. *The Apache Software Foundation: http://tomcat.apache.org/*
2. *DNP3 Organization's Website. http://dnp.org/*
3. *GE Fanuc, Proficy HMI/SCADA- iFIX: http://www.gefanuc.com/en/ProductServices/AutomationSoftware/Hmi_Scada/iFIX/*
4. *Geer, D., "Security of Critical Control Systems Sparks Concern," IEEE Computer, January 2006, Pages 20-23.*
5. *Graham, J.H. and Patel, S.C., "Security Issues in SCADA Systems," DIAS Technology Review, The International Journal of Business and IT, 1(2), 2005.*
6. *Graham, J. H., and Patel, S. C., "Correctness Proofs for SCADA Communication Protocols," Proceedings of World Multi-Conference on Systemics, Cybernetics and Informatics, Orlando, FL, July, 2005.*
7. *iFIX brochure, http://www.geindustrial.com/products/brochures/iFIXDataSheet.pdf http://www.gefanuc.com/en/ProductServices/AutomationSoftware/Hmi_Scada/index.html*
8. *JPC (Java for Process control), NetModule Inc., http://www.netmodule.com/d/produkte/jpc.asp*
9. *National Instrument Corporation, http://www.ni.com/aap/*
10. *Patel, S.C., Tantalean, R. Z., Ralston, P.A., Graham J. H., "Supervisory Control and Data Acquisition Remote Terminal Unit Testbed," Intelligent Systems Research Laboratory Technical Report TR-ISRL-05-01, Dept. of Computer Engineering and Computer Science, University of Louisville, Louisville, February 2005.*
11. *Palu, S. "Java Automation And Related Specification Requests Java Automation" 07/12/2002, http://www.developer.com/java/other/article.php/1402551*
12. *Su, C. -L., Lu, C. -N., and Lin, M. -C. "Wide area network performance study of a distribution management system," International Journal of Electrical Power & Energy Systems, Volume 22, Issue 1, January 2000, Pages 9-14.*
13. *Tantalean-Carrasco, R. Z., "Internet-Based Expert System for Monitoring and Control of Chemical Process," Doctoral Dissertation, Department of Chemical Engineering, University of Louisville, Louisville, December, 2003.*