A

N-Secrecy Authentication Response to Graduated Threat Levels in SCADA Networks

James H. Graham and Waleed Elsaid
Intelligent Systems Research Laboratory
University of Louisville, Louisville, KY 40292
jhgrah01@louisville.edu

Abstract: This paper presents a new approach to authentication for SCADA communications using a computationally light-weight encryption protocol which is capable of resisting most major cyber attacks including distributed, denial of service attacks against a SCADA network. In addition this protocol can be implemented in an extended form which will permit either multiple levels of authentication for various system threat levels, or multiple levels of authentication to support a hierarchy of privileged operations at the RTU.

Keywords: Authentication, symmetric encryption, DNP-3, N-secrecy

1 Introduction

Supervisory control and data acquisition (SCADA) systems monitor, regulate, gather information from, and control the computer-based field units that control much of our industrial and utility infrastructure. Security was not a high priority in the design of much of this equipment, which can be as much as thirty years old, because it used obscure protocols and relatively secure, dedicated communications links. This has all changed in the last twenty years as many SCADA installations are now accessible, either directly or through system design flaws, through the public Internet. Several cyber-based intrusions into SCADA systems have been recently documented [1,2] and terrorist threats against SCADA systems have been reported [3]. Several recent papers have discussed the many threats and security issues in current SCADA networks [4-11].

The Intelligent Systems Research Laboratory (ISRL) at the University of Louisville has an on-going project funded by the Dept. of Homeland Security through the Kentucky Critical Infrastructure Protection Institute to investigate improved methods for detecting, and mitigating, intrusion threats in SCADA systems in chemical plants and other critical infrastructure. This project has three thrusts: (1) distributed correlation filtering for detection of cyber intrusions into SCADA networks, (2) hardening approaches for SCADA remote terminal units (RTUs) to make them more resistant to cyber attack, and (3) approaches to improve authentication on the communications links used by SCADA systems. Reasonable progress has been made during 2006 on all three components of this research effort, and during 2007 experimental evaluations of these results on an SCADA test-bed will be performed. This paper reports on new a new direction in authentication which is being pursued at the ISRL at the University of Louisville to allow for multiple levels of authentication in response to the perceived threat level of the network. The second section of this paper gives a brief overview of communications security issues in SCADA and reviews some security enhancement options. The third section gives the details of the new N-secrecy approach to respond to graduated threats. The fourth section presents conclusions and future directions for this effort.

2 Communication Security in SCADA Systems

As discussed in the previous section, communications security was not given a high priority in the design of most of the existing SCADA deployments because it used obscure protocols and relatively secure, dedicated communications links. This has changed as newer protocols and designs have attempted to take advantage of the lower costs and greater accessibility offered by the ubiquitous TCP/IP protocols used by both the public Internet and many private and proprietary networks as well. In particular, the Distributed Network Protocol (DNP) was designed to ride on top of TCP/IP and is now widely used in SCADA and other distributed control system (DCS) implementations. Two methods of securing SCADA communication have been considered: (1) wrapping SCADA protocols with external cryptographic protocols, and (2) enhance SCADA protocols with cryptography techniques. Both of these will be briefly examined in the following paragraphs.

2.1 Wrapping to Enhance Communications Security

We have investigated two possible wrapping options: (1) use the Secure Sockets Layer (SSL)/Transport Layer Security (TLS) protocol and (2) use the IPsec protocol. SSL/TLS protocol was designed to protect against an intruder who may modify, delete, or replay the data in transit. This protocol is commonly used to protect client/server communication over the Internet from threats of eavesdropping, tampering, and message forgery. The protocol allows authentication of sending and receiving devices and provides integrity by using digital signatures and privacy using encryption. SSL/TLS is administered by an international standards organization (IETF) and is well established in areas of web browsers, web servers, and other Internet systems. The SSL/TLS solution has some limitations as it can run only a reliable transport protocol such as TCP. Second, this protocol relies on other components such as encryption and signature algorithms; therefore, SSL/TLS implementation cannot be stronger than the cryptographic or signature tools on which this implementation is based.

The other “wrapper” option is using IPsec (secure IP). SSL/TLS operates above the Transport Layer (OSI Layer 4) but IPsec operates at the Network Layer (OSI Layer 3). IPsec can prevent attacks launched by repeatedly breaking connections because these connections are established at the IP level. However, IPsec cannot provide application-specific security or advanced authentication security features. Moreover, IPsec is more sensitive than SSL/TLS to interference by intermediaries in the communication channel.

So it is would be cumbersome to send encrypted or authenticated data to a computer behind a firewall when IPsec is used.

2.2 Using Selected Cryptography Techniques to Enhance SCADA Protocols

SSL/TLS and IPsec provide generic security features and protect only a part of the communication channel, but selective cryptography techniques can provide end-to-end security that protects an entire message between a sender and the receiver. Based on early works of the DNP-3 User Group, we have implemented and tested the following two cryptographic-based enhancement techniques the University of Louisville.

2.2.1 Authentication Octets

This technique is based on a digital-signature algorithm. In this technique, additional bytes of information, referred here as Authentication Octets, are appended to each message from an MTU to an RTU for the authentication. First, hash is performed on selected bytes of a message, resulting in a hash digest. This digest is encrypted by using the sender's private key and then the encrypted digest is sent along with the message. In our test-bed implementation, the MTU stored all the public keys, eliminating the need of the Certificate Authority. The message itself is not encrypted to save the processing time during the encryption and the decryption. Other data that vary with time are time stamped for additional checks. For example, with the timestamp, the receiver verifies that the time of reception does not vary from the time of transmission beyond a specified range to eliminate an intruder from resending a valid message multiple times. The receiver decrypts the hash digest by using the sender's public key and compares it with hash digest independently calculated by the receiver. The receiver would conclude that the message came from an authentic source if it can decrypt the digest using the sender's private key. The receiver would also conclude that the message contents are unaltered if the digest values match.

The Authentication Octets technique may also be applied to the messages sent from an RTU to an MTU to prevent an intruder from making an MTU send inappropriate messages. For example, an intruder can send valid looking RTU values, indicating a high voltage surge to prompt an MTU to send a control message to turn-on the circuit breaker.

The Authentication Octets technique protects spoofing and modification attacks. Because the entire message is not encrypted, this technique does not protect from the threat of eavesdropping. However, for SCADA networks, eavesdropping is not a major concern. The more significant concern is to block the unauthorized commands to the RTU.

2.2.2 Authentication via Challenge Response

This technique verifies the identity of an RTU or an MTU by using the challenge-response cryptography to protect against a man-in-the-middle attack. Either of the devices (MTU or RTU) initiates a challenge to authenticate the other device that shares a secret. The authenticator device sends a random challenge to the other device. Using the secret and the challenge the other device calculates a hash digest and then sends this digest as a response. The authenticator checks the response against its own calculation of the expected hash digest. If the values match, the authentication is acknowledged, otherwise the authenticator terminates the connection. Whenever an authentication is required, a new and different challenge is sent, repeating the above steps. More details about these two enhancement approaches, including formal correctness proofs, can be found in [12].

3 The N-Secrecy Approach

The N-Secrecy approach takes a more pervasive approach to user and message authentication in SCADA networks. The goal is to develop light-weight protocols using symmetric encryption, where possible, that allows multiple levels of authentication which can be adjusted to meet threat conditions on the network. In the discussion which follows we first introduce the idea of double secrecy authentication and explain its benefits, and we then extend this approach to the multiple levels of authentication.

3.1 Double-Secrecy Authentication

The idea is to keep two private keys, two secrets, and a hash table at each unit,

1. One private key between the Master Terminal Unit (MTU), and all the Remote Terminal Unit (RTUs), that can communicate with, general purpose private key (GPK).
2. Another one between the MTU and each RTU, a specific private key (SPK).
3. Two secrets between the MTU and each RTU, or each pair of sender-receiver (SSEC1 and SSEC2)
4. A hash table containing Key-Value pairs list that corresponds to a list of SSEC1(SPK, SSEC2), i.e. Key is SSEC1, and value is (SPK, SSEC2).

To send a message from the MTU to the RTU or vice versa, the following authentication scenario happens, (see Figure 1).

1. The header of the message contains the secret shared between sender and receiver (SSEC1) encrypted using the GPK.
2. The message body contains the actual message encrypted using the SPK.
3. The footer of the message contains SSEC2
4. The message is sent to the receiver.
5. The receiver decrypts the header using the GPK.
6. The receiver uses its own hash table to look up the message header.
7. If the hash-table lookup process results in Null, this would mean an authentication failure. Since, this means that the encrypted script wasn't SSEC1 at the first place.

8. If the hash-table lookup process succeeds, this will mean that the header is SSEC1, which is then used to get the SPK, and SSEC2.
9. The SPK is then used to decrypt the footer of the message.
10. If the value of the decryption equals the SSEC2, then the message is authenticated.
11. SPK is then used to decrypt the message.

From the above discussion, we note that, unlike SSl/TLS, this model does not use public key cryptograph. Using irreversible functions to encrypt data requires high intensive computation, which is not desirable, and at the same time, why would we use public keys in a private network anyway? Secondly, the protocol depends on two levels of the security which makes very secured, as well as lightweight.

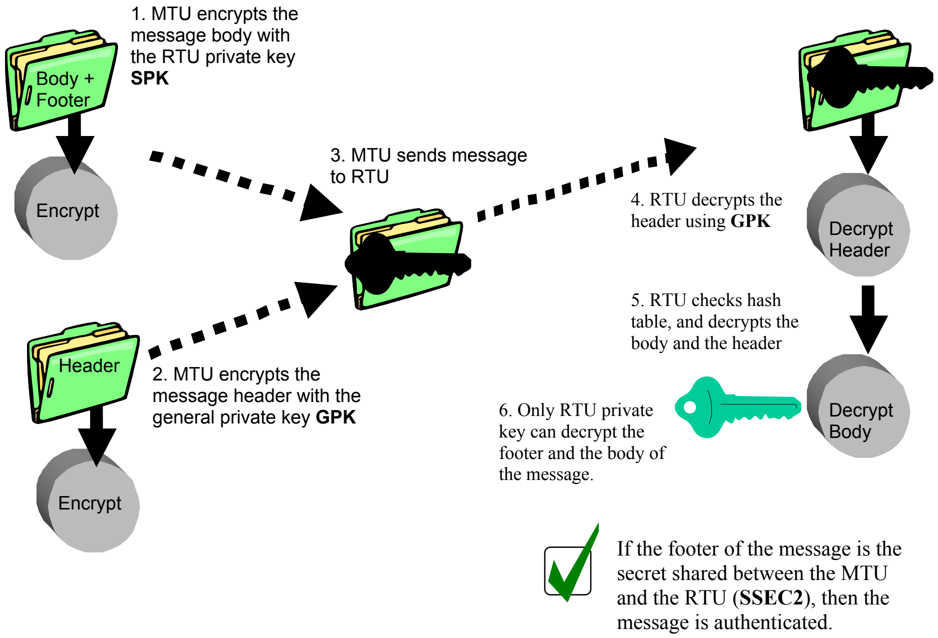
1. MTU encrypts the message body with the RTU private key **SPK**
Body + Footer
Encrypt
3. MTU sends message to RTU
4. RTU decrypts the header using **GPK**
Decrypt Header
5. RTU checks hash table, and decrypts the body and the header
Header
2. MTU encrypts the message header with the general private key **GPK**
Encrypt
6. Only RTU private key can decrypt the footer and the body of the message.
Decrypt Body
If the footer of the message is the secret shared between the MTU and the RTU (**SSEC2**), then the message is authenticated.

Figure 1 – Double Secrecy Protocol

3.2 Analysis of Double Secrecy Protocol for Preventing Denial of Service Attacks

The objective of this part is to check out how the double secrecy authentication pattern for protocol security, can withstand Denial of Service (DoS) attacks. DoS attacks occur when many bogus messages are sent to the terminal unit, which has no problem in identifying that those are bogus messages because the authentication technique used, but the DoS happens when the bogus messages are sent to the terminal unit in a large number in a very short time. The terminal unit would spend too much time processing the authentication technique before it figures out that they are bogus, after a while the unit becomes overloaded, and it denies the service to authentic messages.

The classic solution to this problem is simply to use some kind of a proxy or filter to block out the IP causing those fake messages. Now the real problem comes with using a Spoofing-DOS attack, that is when some intruder gets inside the system of a real terminal, and sends the fake messages, because at this time the IP blocking will simply blocks the real terminal that is spoofed, and whenever the spoofing ends and the spoofed terminal becomes in its own again and tries to send a real messages it cannot deliver the message because the IP is blocked. Hence, a more practical solution is needed to handle the DOS attack.

In the Double-Secrecy, the key point is that there are two tests for the message to be authenticated: (1) first the header of the message should be a key in the receiver terminal hash-table after being decrypted by the GPK, and second, the footer should be the SSEC2 (from the hash-table) after being decrypted by the SPK. A small loss of the processing time occurs because the receiving unit should wait for the whole message to come and process the first test to know it's a bogus message, or a potentially authenticate message. As shown in Figure 2, we use an inspector component to take the first packet of the header when it comes and perform some pattern matching to check if it's promising to be a part of some key in the hash-table. That way, bogus messages can be detected from the very first packet. Whenever, a legitimate message header is detected, it will be forwarded to the terminal to proceed with the two tests.

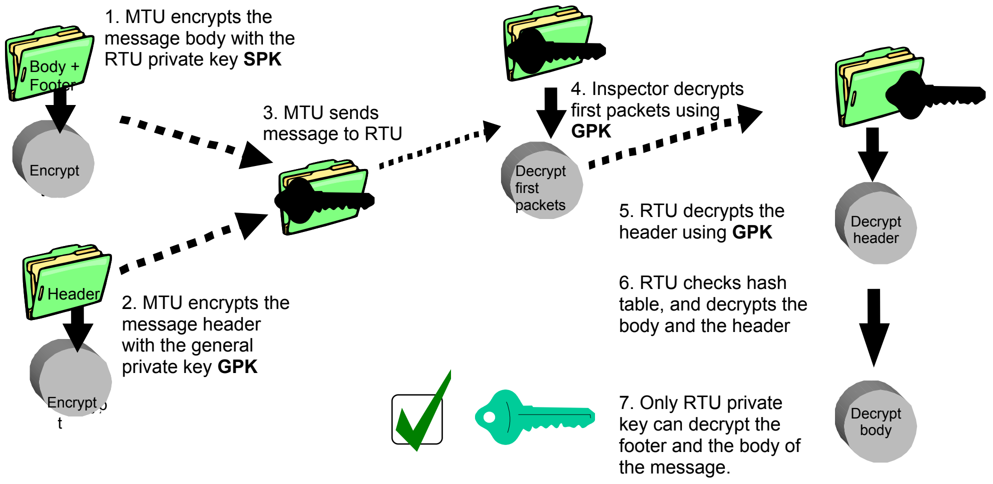
Body + Footer
1. MTU encrypts the message body with the RTU private key **SPK**
Encrypt
3. MTU sends message to RTU
4. Inspector decrypts first packets using **GPK**
Decrypt first packets
5. RTU decrypts the header using **GPK**
Decrypt header
Header
2. MTU encrypts the message header with the general private key **GPK**
Encrypt
6. RTU checks hash table, and decrypts the body and the header
7. Only RTU private key can decrypt the footer and the body of the message.
Decrypt body

Figure 2 – Double Secrecy with DOS Suppression

3.3 N-Secrecy Authentication Approach

The objective of this section is to make the level of security on the SCADA systems protocols on demand, that is initialized it on level L and increasing to L2 in case of sending very vital messages, or detecting an increase of the threat level, and further increase it, say to level LN, in case of being under severe attack. Loosely speaking, this creates states of security like now-familiar green, yellow, orange, and red security levels used by DHS and now posted in all US airports.

The idea here is to generalize the double secrecy authentication framework to contain N+1 secrets instead of two. The exact scenario will be as follows:

1. The header of the message contains the secret shared between sender and the receiver (SSEC0) encrypted using the GPK.
2. The footer of the message contains n secrets; n>0.
3. The message body contains the actual message encrypted using the SPKn.
4. The message is sent to the receiver.
5. The receiver decrypts the header using the GPK.
6. The receiver uses its own hash table to look up the message header.
7. If the hash-table lookup process results in Null, this would mean an authentication failure.
8. If the hash-table lookup process succeeds, this will mean that the header is SSEC1, which is then used as the key in the hash-table to give a vector of pairs (SPKi, SSECi); i=1..n.
9. The length of those secrets is known for both the sender and the receiver.
10. An iteration is being done to read the SSECi length number of bytes, decrypt them using SPKi, and compare it to the SSECi.
11. If comparison succeeds, then i is incremented, to read the next one.
12. If comparison fails at any stage, then the sender is not authenticated.
13. SPKn is then used to decrypt the message.

The N-Secrecy approach can be used both for achieving multiple levels of authentication, in response to heightened threat conditions, and to segment messages according to criticality. In this approach, each remote terminal unit will be organized in such a way that, there is a master component, and a vector of operation components, categorized in a hierarchical manner. For example read data points operations would likely be at a low security level, while write operations would reside at higher levels. For every node there will be a corresponding authentication and authorization attributes, and for every child node the attributes will be accumulative from the ancestor. The overall scenario will be that the first secret which is SSEC0, will contain some information about the type of operation, so that the master component in the terminal unit will forward the message to the right operator component. As more secrets are read, through the N-Secrecy framework, and the corresponding tests are passed, the more information the operator component knows about the kind of operation the sender needs, allowing for a dynamic authentication process.

4 Conclusions and Future Directions

Security of SCADA communications will likely continue to be a significant practical and research concern for many years. Potentially there are some hardware enhancements which will significantly improve the resistance of SCADA RTU's against cyber attacks. But this will leave a legacy of older equipment which represents a large capital investment, and will be replaced at a very slow rate under upgrade and replacement policies. The best bet for fortifying these older, legacy systems would seem to be enhanced authentication protocols, possibly including some of those discussed in this paper. While our work has focused specifically on DNP3, it should also be equally applicable to Modbus and IEC protocols.

There are some remaining issues with the N-secrecy authentication protocol, particularly with respect to heavy DDoS attacks. The inspector component itself can get loaded and the same problem of loading and then being out of service happens with the component, which will then lead to DoS state of the whole unit. We can use a vector of inspectors with some sort of a master inspector component that act like a load balancer, in a sense that whenever, some inspector component becomes almost loaded, the master inspector will create another inspector component from the pool of inspectors available, and gets it in service. If it happens that all of the inspectors in the pool are in service, it will simply block this IP for a short time until one becomes free. Another potential major problem is that encryption/decryption functions need not be linear, in fact almost all of them are not, in a sense that "F(pq)!=F(p)F(q)", which means that even if the message is not authentic this doesn't mean that the decryption of the first packet of the header would be the first packet of the decrypted header. We can overcome this problem by reconstructing the key in the hash-table.

About the Authors - Dr. James H. Graham is the Henry Vogt Professor of Computer Science and Engineering at the University of Louisville. He holds a joint appointment in the Department of Computer Engineering and Computer Science and the Department of Electrical and Computer Engineering. He has previously served as a faculty member at Rensselaer Polytechnic Institute, as a visiting professor at Purdue University, and as a design engineer with General Motors Corporation. He has served as a technical consultant to the General Electric Research and Development Center, GTE Laboratories (now part of Verizon), Science and Engineering Services, Inc., and several other high technology companies. He earned the Bachelor of Science degree in electrical engineering from the Rose-Hulman Institute of Technology in 1972, and the Master of Science and Ph.D. degrees in electrical engineering from Purdue University in 1978 and 1980, respectively. Dr. Graham is a senior member of the Institute of Electrical and Electronic Engineers, a member of the Association for Computing Machinery, a member of ISCA and a registered professional engineer. He is a cofounder and the current director of the Intelligent Systems Research Laboratory in the J. B. Speed School of Engineering at the University of

Louisville. Dr. Graham's research interests include robotics, artificial intelligence, high-performance computing, computer simulation, and computer security.

Mr. Waleed Elsaid has a Master's Degree in Computer Science from Ain Shams University in Egypt. He is currently a PhD student in the Computer Science and Engineering Program at the University of Louisville and a Graduate Research Assistant in the Intelligent Systems Research Laboratory.

References

[1] E. Byres, J. Lowe, "The Myths and Facts behind Cyber Security Risks for Industrial Control Systems." VDE Congress, Berlin, October 2004.

[2] NRC, "Potential Vulnerability of Plant Computer Network to Worm Infection," U.S. Nuclear Regulatory Commission, Notice 2003-14, Washington, DC, 2003.

[3] K. Poulsen, "Brits found OpenSSL bugs," *SecurityFocus,* September, 2003.

[4] V.M. Igure, S.A. Laughter, R.D. Williams, "Security Issues in SCADA Networks," Computers and Security, 2006, www.elsevier.com/locate/cose

[5] P. Ralston, J. Graham, S. Patel, "Literature Review of Security and Risk Assessment of SCADA and DCS Systems," Technical Report, TR-ISRL-06-01, Intelligent Systems Research Laboratory, University of Louisville, 2006.

[6] D. Peterson, "Securing IP Control Protocols," Control Magazine, 2005, www.controlglobal.com

[7] B. Singer, J. Weiss, "Control Systems Cyber Security," Control Engineering, 2005.

[8] J. Falco, K. Stouffer, A. Wavering, A., F. Proctor, "IT Security for Industrial Control Systems," Intelligent Systems Division, National Institute of Standards and Technology (NIST) Gaithersburg, MD, in coordination with the Process Control Security Requirements Forum(PCSRF), 2006.

[9] DOE, "21 Steps to Improve Cyber Security of SCADA Networks," President's Critical Infrastructure Protection Board and Department of Energy Report, Washington, DC, 2002.

[10] D. Peterson, "Intrusion Detection and Cyber Security Monitoring of SCADA and DCS Networks," ISA Automation West, ISA, 2004.

[11] AGA, "Cryptographic Protection of SCADA Communications: Retrofitting Serial Communications," AGA Report 12-1, American Gas Association, 2006.

[12] S. Patel, "Secure Internet-based Communications Protocol for SCADA Networks," Doctoral Dissertation, University of Louisville, 2006.