SCADA System Security, Complexity, and Security Proof

Reda Shbib, Shikun Zhou, Khalil Alkadhimi

School of Engineering, University of Portsmouth, Portsmouth, UK
{reda.shbib,shikun.zhou,khalil.alkadhimi}@port.ac.uk

Abstract. Modern Critical infrastructures have command and control systems. These command and control systems are commonly called supervisory control and data acquisition (SCADA). In the past, SCADA system has a closed operational environment, so these systems were designed without security functionality. Nowadays, as a demand for connecting the SCADA system to the open network growths, the study of SCADA system security is an issue. A key-management scheme is critical for securing SCADA communications. Numerous key-management structures for SCADA also have been suggested. 11770-2 Mechanism 9 Key establishment Protocol has been used in SCADA communication however a security proof for the 11770-2 Mechanism 9 protocol is needed. The purpose of this paper is to provide a general overview about SCADA system, and its related security issues. Furthermore, we try to investigate the importance of key management protocol and the need of formal security poof.

Keywords: SCADA, key management, 11770-2 Mechanism 9, Formal security poof.

1 Introduction

SCADA systems are used to control and monitor assets where central data acquisition is as important as control[1, 2] .These systems are used in distribution systems such as water distribution and wastewater collection systems, oil and gas pipelines, electrical utility transmission and distribution systems, and rail and other public transportation systems. SCADA systems integrate data acquisition systems with data transmission systems and HMI software to provide a centralized monitoring and control system for numerous process inputs and outputs. SCADA systems are designed to collect field information, transfer it to a central computer facility, and display the information to the operator graphically or textually, thereby allowing the operator to monitor or control an entire system from a central location in real time. Based on the sophistication and setup of the individual system, control of any individual system, operation, or task can be automatic, or it can be performed by operator commands.[1, 3]

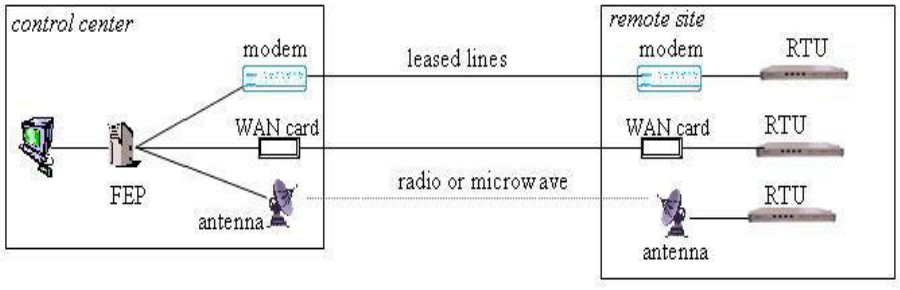

Fig. 1. Simple SCADA System

2 SCADA vulnerabilities

2.1 SCADA System Vulnerabilities

Critical infrastructures are facing important threats as the development in the use of SCADA systems and the integrated networks. In addition, the complicated infrastructure offers huge capabilities for operation, control, and analysis; it also increases the security risks due to cyber vulnerabilities The Development of SCADA systems have also attracted some issues about cyber-attacks. The SCADA industry is transitioning from a legacy environment, in which systems were isolated from the Internet and focused on reliability instead of security, to a modern environment where networks are being leveraged to help improve efficiency. In addition the connectivity of SCADA networks with outside networks will continue to grow, leading to an increasing risk of attacks and the significant need to advance the security of these networks [4, 5]

Furthermore, open communication protocols such as Modbus and DNP3 are increasingly used to achieve interoperability, exposing SCADA systems to the same vulnerabilities that threaten general purpose IT systems[6]. The integration of SCADA networks with other networks has made SCADA vulnerable to various threats [5]. Many SCADA protocols use TCP/IP (Transmission Control Protocol/Internet Protocol) and provide no additional protection. Vulnerabilities in the TCP/IP protocol include IP spoofing and man-in-the middle attacks. Additionally, the standardization of software and hardware used in SCADA systems potentially makes it easier to mount SCADA-specific attacks, as was evident in thecae of Stuxnet. Stuxnet was a piece of malware created to specifically target control systems [6, 7]

2.2 Security Concerns

Most of Countries are becoming significantly reliant on automated Supervisory Control and Data Acquisition (SCADA) systems to support deliver critical services. SCADA systems, which once used proprietary communication mechanisms, are using standard protocols, such as DNP3 [8]. Security incidents have significantly increased

since 1988 and recorded by CERT to 137.529 incidents in 2003 (CERT/CC Statistics 1988- 2005).The necessity to make SCADA systems more secure has therefore been classified as a significant field of research. One of the most important security requirements for SCADA systems is that communication channels must be more secured. Secure keys need to be established before cryptographic techniques can be used to secure communications [9].

3 Formal Security Proof

3.1 Security Proofs

Security proofs were major concerns of many of research in last few years. To ensure that a protocol or a software have a certain requested properties, is an important issue. This task has to be done by formal reasoning instead of examinations and simulations, as the latter approach is not as comprehensive as the formal one.[10]

Security proofs are methods to validate the security of a protocol. A reductionist security proof of a protocol helps to show that the security in the proof model is related to the cryptographic primitives used. A security proof attempt to show that a protocol meets the defined goals for the protocol in the security model used.[11]

3.2 Protocols Verification Approaches

Protocol verification has mainly two possible approaches: The formal model and the computational model

- In the first model, we are in a very idealized setting; thus this can be efficiently implemented in completely automated protocol verifiers.
- The second model inspires ideas from complexity theory and needs more human interference in proofs, and it is being automated only in very recent times. [12]

These verification methods let us to discover and uncover design faults that can stay hidden for years. The purpose of the present paper is to investigate a proof of security on the 11770-2 Mechanism 9 protocol in the formal model.

The purpose of this protocol 11770-2 mechanisms 9 (ISO 1996) is to establish a long-term key shared between the nodes.

4 Protocol Description

The ISO 11770-2 standard has been published in 1996, and specifies a series of protocols for establishing shared secret keys using symmetric cryptographic techniques. The protocols in this standard use a many of different mechanism in order to ensure the freshness of the established keys, and offer several cryptographic assurances techniques of the established keys. [13]

We are mainly concerned about ISO 11770-2 mechanism 9 which used as a basis for the node-node key establishment protocol. This mechanism has chosen, as it is the best fit for SCADA systems. In the case of SCADA, it is more appropriate for the generation of keys to be performed by the external device, and not have keys generated by the nodes in the systems

1. $B \rightarrow A:$ N_B
2. $A \rightarrow S:$ N_A, N_B, B
3. $S \rightarrow A:$ $\{N_A, K_{AB}, B, Text1\}_{K_{AS}}$
   $\{N_B, K_{AB}, A, Text2\}_{K_{BS}}$
4. $A \rightarrow B:$ $\{N_B, K_{AB}, A, Text2\}_{K_{BS}}$
   $\{N'_A, N_B, B, Text3\}_{K_{AB}}$
5. $B \rightarrow A:$ $\{N_B, N'_A, Text4\}_{K_{AB}}$

Figure 1: ISO 11770-2 Mechanism 9

- A and B are nodes that need to establish a key
- NA is a nonce that is created by node A.
- NA′ is a second nonce created by node A.
- S is the server (representing the Key Distribution Centre)
- A → B is message sent from node A to node B
- KAB is the shared key between node A and B
- {Text} KAB is the encryption of the message text, using the key KAB.
- Strings Text1 to Text4 are text messages

5 Security Proof

The fact that a security proof depends on the model used. The security model will outline the aims for security, and the controls given to the opponent. The selection of the accurate model has the impact on the value of the security proof. In this section we try to investigate a security proof of 11770-2 mechanisms 9 protocol, it is assumed to use Bellare Rogaway model .Bellare Rogaway model has been developed and refined over many years, with some different versions, which has been used for different proofs. It is very important in any security proof to specify the adversarial model and a clear definition of security. We will follow in our developing the proof of Boyd, Choo and Mathuria .

5.1 Reductionist Security Proofs

Reductionist security proofs are a significant part of generating valuable and safe cryptographic protocols. The aim of provable security is to demonstrate that a protocol will meet the security goals .The final outcome will be to say that breaking any of

security properties will require an attacker to have broken a fundamental security primitive. [14].

Several number models for security proofs performance of protocols have been proposed. The model that will be used as the starting point for the security proof of 11770-2 mechanisms 9, is the Bellare Rogaway model.

Cryptographic community is essential to validate security proofs. Many proofs have been found to have flaws, and it is necessary to be validated before it is established. Although the use of provable security techniques is not perfect, and does not promise a full security, they do offer important tools for helping to validate the security of a protocol [15]

6 Proposed Bellare Rogaway Model

Bellare Rogaway model permits the development of reductionist security proofs in order to validate that the desired protocol is secure, meeting the specified goals. The reductionist security proof will demonstrate that in order to breakdown the protocol, an adversary must attack the encryption function, which the protocol depends on. Thus by implementing the protocol using strong encryption algorithm the protocol will be more secure.

The security is defined as the advantage of the adversary in distinguishing session keys from random strings and is used to define the security of the protocol. Furthermore, a secure protocol must complete successfully with the principals accepting the session key if the adversary does nothing to interrupt the protocol.[16]

The Bellare Rogaway model has a strict definition of security (in distinguish ability of established keys from random keys). Protocol is supposed to be insecure if it employs the new key to encrypt any messages

As the entity authentication messages that form a part of the 11770-2 protocol use the established key, it will be classified as insecure in the Bellare Rogaway model, since the adversary can check whether authentication works for the string it has been given, if the string is the session key then the authentication check will work but if the string is a random string then the authentication will fail.[16]

In this section we have introduced the Bellare Rogaway model for developing reductionist security proofs. This model is a well-established approach for verifying security. We are going to use Bellare Rogaway model to create a reductionist security proof, which proves the security of the key establishment protocol 11770-2

Conclusion

This paper provided a detailed discussion on critical infrastructures and the role cryptographic mechanism protocol plays in their protection. We examined one of current protocol 11770-2 mechanisms and determined that the solutions are not sufficient for such an interconnected infrastructure. We provided our initial framework design for the security proofing of this protocol following the ‘model’ that we will use in the future.

Our future work will focus on the introducing the Bellare Rogaway Model for developing a reductionist security proofs. This model has been well established for verifying security .Next step is to show how the security goals in this model meet the set of security aims of 11770-2 mechanisms 9. We will use Bellare Rogaway model to create a reductionist proof which proves the security of the key establishment protocol.

References

1. C. Ning, et al., "SCADA system security: Complexity, history and new developments," in Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on, 2008, pp. 569-574.
2. S. Gold, "The SCADA challenge: securing critical infrastructure," Network Security, vol. 2009, pp. 18-20, 2009.
3. R. E. Johnson, "Survey of SCADA security challenges and potential attack vectors," in Internet Technology and Secured Transactions (ICITST), 2010 International Conference for, 2010, pp. 1-5.
4 C. Donghyun, et al., "Efficient Secure Group Communications for SCADA," Power Delivery, IEEE Transactions on, vol. 25, pp. 714-722, 2010.
5. S. Rautmare, "SCADA system security: Challenges and recommendations," in India Conference (INDICON), 2011 Annual IEEE, 2011, pp. 1-4.
6. K. Stouffer , et al., "Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security " The National Institute of Standards and Technology (NIST), 2006.
7. C. Office, "The UK Cyber Security Strategy Protecting and promoting the UK in a digital world," ed, 2011.
8 R. Brewer, "Protecting critical control systems," Network Security, vol. 2012, pp. 7-10, 2012.
9. V. M. Igure, et al., "Security issues in SCADA networks," Computers & Security, vol. 25, pp. 498-506, 2006.
10. J. Goubault-Larrecq, "Towards Producing Formally Checkable Security Proofs, Automatically," in Computer Security Foundations Symposium, 2008. CSF '08. IEEE 21st, 2008, pp. 224-238.
11. A. Carcano, et al., "Scada Malware, a Proof of Concept," in Critical Information Infrastructures Security. vol. 5508, R. Setola and S. Geretshuber, Eds., ed Berlin: Springer-Verlag Berlin, 2009, pp. 211-222.
12. R. Bresciani and A. Butterfield, "A formal security proof for the ZRTP Protocol," in Internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference for, 2009, pp. 1-6.
13. ISO, "Information technology — Security techniques — Key management " ISO/IEC, 2008.
14. A. Stolbunov "Reductionist Security Arguments for Public-KeyCryptographic Schemes Based on Group Action," presented at the NISK, 2009.
15. N. Koblitz, "Another Look of "Provable Security"," Journal of Cryptography, vol. 20, p. 37, 2007.
16. R. Dawson "Secure Scada Communication " 2008.