Security Enhancement Mechanism of Modbus TCP Protocol

Xuan Luo and Yongzhong Li

ABSTRACT

Considering the security problem of the Modbus TCP protocol, this paper proposes a secure protocol, Modbus-E. It uses symmetric key and digital signature technology to ensure the confidentiality and authentication of data. It also uses the synchronization principle and the mono-direction principle of the hash function to ensure the uniqueness of data. Through the Filtering method of "white list", it can guarantee the controllability of instruction, ultimately without any increase in communication process to achieve secure communication. Through the verification and analysis of experiment, Modbus-E protocol can prevent the authentication attack, man-in-the-middle attack and replay attack of the instruction by the attacker. Compared with existing methods, this method is more secure and can comprehensively improve the security of Modbus TCP communication.¹

INTRODUCTION

Industrial control systems are initially closed and small in scale, and only need to ensure physical security to basically meet the requirements, so the initial design of field bus protocol Modbus did not take security factors into account. With the development of industrial control systems and the wide application of TCP IP protocol, Modbus protocol is integrated with TCP IP to form Modbus TCP[1-2]. However, Modbus TCP fails to solve the security problems existing in the original Modbus protocol, and even introduces the design defect of TCP IP protocol, which makes the security problem of Modbus TCP protocol more serious[3-4].

¹Xuan Luo, Yongzhong Li, College of Computer, Jiangsu University of Science and Technology, Zhenjiang Jiangsu, China
In view of the security problems of Modbus TCP, Yang Jing[5] designs as security module based on trusted computing, but the data processing is time-consuming and does not meet the real-time requirements. Fovino[6] proposed adding a timestamp and a hash chain field to the Modbus protocol and encrypting it with a private key, but the data is still transmitted in plain text, some sensitive data is easily intercepted by attackers, and there is still a problem of functional code abuse. Hayes[7] proposed a hash-based streaming control protocol STCP while the security of the scheme is based on pre-shared key, once the pre-shared key is cracked, the security will be destroyed.

In view of the problems in the above references, this paper makes in-depth research on the abuse of function code, secure data transmission and feasibility of the scheme of Modbus TCP protocol. This paper improves the Modbus protocol itself and designs a Security Enhancement protocol, Modbus-E protocol. The protocol can resist replay attack, authenticate both sides of sending data, encrypt key fields of data, and prevent abuse of function codes by "white list" method. Therefore, the safety of the industrial control network can be fundamentally improved.

MODBUS TCP SECURITY ANALYSIS

The Modbus TCP protocol application data unit ADU is composed of a 7-byte protocol header MBAP and an N-byte protocol data unit PDU, its message format is shown in Figure 1.

Modbus TCP protocol design does not consider safety factors, and its security risks can be listed as follows:

Lack of authentication: Attackers can impersonate both sides to send malicious instructions, such as the virtue of Stuxnet that once shocked the world, hackers attack industrial control system by cheating, which resulting in serious damage to the system.

Function code abuse: Anyone who has access to the terminal can send arbitrary instructions, the server has no corresponding processing mechanism.

Data transmission in plain text: Hackers can easily intercept the data packet and even tamper with the data message to turn it into malicious message, in order to damage the industrial control system.

<table>
<thead>
<tr>
<th>TransmissionID</th>
<th>Protocol</th>
<th>Length</th>
<th>UnitID</th>
<th>Function</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>MBAP</td>
<td>PDU</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1. Modbus TCP message format.
MODBUS-E PROTOCOL'S DESIGN

Modbus-E Protocol's Format

In order to realize authentication, prevent tampering, anti-replay and other mechanisms, and to ensure the availability of the protocol for industrial control system, this paper directly modifies the Modbus TCP protocol without adding communication process, and proposes a secure Modbus TCP protocol to realize the data security of industrial control system. The format of the Modbus-E protocol is shown in Figure 2.

Synchronization Mechanism Based on Hash Chain

Choose a random number as seed, calculate its hash value, and use its first byte data as synchronization identifier code. After each match is successful, the original hash value is hashed to calculate the next synchronization identifier sent, and then update each synchronization identifier code by analogy. Utilizing the unidirectional principle of hashing to ensure the uniqueness of the data, which can effectively resist replay attacks. At the same time, there is no complete hash value in the packets, which greatly increases the difficulty of the synchronization identifier being cracked. The seeds of synchronous identification can be updated periodically, and the seeds of both sides can be synchronized by asymmetric key algorithm.

"White List" Filtering Mechanism

The design idea of "White List" is based on the four key factors of "Level-IP Address-Device ID-Function Code" to implement rule filtering. Its rule model settings are shown in Figure 3, the filtering process is shown in Figure 4.

The Level field can be defined as different values as required, such as 100, 50 and 10, corresponding to roles root, admin and monitor, respectively. Root has the highest authority and can execute all operation commands. Admin can only execute read-write coil and register values, but cannot execute high-risk commands such as restart and shutdown commands. Monitor can only execute read-write coil and register commands. It can also be subdivided according to the actual situation of users. Each identity is bound to an IP. To prevent ARP attacks, the MAC addresses
can be bound on access layer network devices. The white list is stored in a structured array. By defining the structure array in advance and setting the parameters of each structure, when the server receives the data, it will find the structure array with the value of level field, determine the structure to which it belongs, and then match the corresponding parameter values in turn. If any attribute does not match, it will report an error and discard the data packet. The establishment of a white list based on role can effectively prevent the abuse of functional codes.

<table>
<thead>
<tr>
<th>Level</th>
<th>IP Address</th>
<th>Device ID</th>
<th>Function Code</th>
</tr>
</thead>
</table>

Figure 3. "White list" rule model.

![White list filtering process diagram]

Figure 4. "Whitelist" filtering process.

![Modbus-E protocol system implementation principle diagram]

Figure 5. Modbus-E protocol system implementation principle.

![Experimental environment diagram]

Figure 6. Experimental environment diagram.
MODBUS-E'S IMPLEMENTATION

Modbus-E Protocol System Design

This protocol system has developed a pair of communication software Modbus-E Client and Modbus-E Server to realize the conversion of Modbus TCP protocol to Modbus-E protocol. The principle of implementation is shown in Figure 5.

Modbus-E Protocol Data Processing

The data processing process of the Modbus-E protocol is as follows:

1. Modbus Client creates an effective Modbus request packet ($M_{req}$);
2. The Modbus-E Client receives the data packet and reconstructs the data packet according to the synchronization identifier ($Syn$), the privilege level ($Level$), the transmission identifier ($TID$), the protocol flag ($PID$), the length field ($L$), and the unit identification field ($UID$), total length field ($Lt$), and PDU field encrypted with AES algorithm key $P_{AES}$, construct N-Modbus field as shown in Figure 2, calculate its hash value by Hash algorithm, sign it with private key ($P_{ri}$), it is then added to the N-Modbus and sent to the modbus-E Server side.

$$N\text{-Modbus} = Syn|Level|TID|PID|L|UID|Lt|\{PDU\}P_{AES} \quad (1)$$

$$C = [N\text{-Modbus}] \{\text{Hash}(N\text{-Modbus})\}P_{ri} \quad (2)$$

3. When Modbus-E Server receives the data packet, it determines first whether its synchronization identifier $Syn_{sc}$ is identical with the synchronization identifier $Syn_{ss}$ of Server communication, then verifies whether its signature is correct, decrypts the PDU field with the shared key, and finally matches the "white list" list rule. If all match returns the result True, otherwise it returns False.

$$R = \text{Check}(Syn_{sc}, Syn_{ss}) \& \text{Auth}(C) \& \text{WhiteListCheck}(C) \quad (3)$$

4. If the matching result $R$ is True, then restore and forward the original request packet $M_{req}$, otherwise discard and alarm.

5. The response packet processing is consistent with the above.

EXPERIMENTAL ANALYSIS

This experiment uses Windows 7 system virtual machine in VMware, Modbus Poll/Slave simulation software and self-developed communication software Modbus-E Client/Server to simulate. The experimental environment is shown in Figure 6.
PC1, PC2, PC3 and PC4 are all client-side. Modbus Poll client simulation software and secure communication software Modbus-E Client are all installed. IP of PC1, PC2 and PC3 are added to the "white list" list. PC5 serves as a server and deploys Modbus Slave server-side simulation software and Modbus-E server-side communication software. Modbus-E protocol is used to communicate between PC1-5 and original Modbus TCP is used to communicate between PC5 and security gateway.

Modbus-E Security Analysis

- **Replay attack:** An attacker replays intercepted data packets, which are discarded due to mismatch of identification codes.
- **Tampering attack:** Attackers send malicious instructions by tampering with intercepted data packets, but because the key fields have been encrypted and the entire field has been hashed and signed, the tampered data packets will lead to the failure of hash value checking and attack failure.
- **Eavesdropping attack:** because the key field has been encrypted, the attacker cannot obtain the key and cannot get the plaintext data.

The results of test show that "white list" can effectively filter illegal operation instructions. This experiment uses only three roles to manage function codes based on roles, and the definition of function codes is relatively simple, so it can be divided in more detail according to the actual needs of users.

### TABLE I. PROTOCOL SECURITY ANALYSIS.

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Document 5</th>
<th>Document 6</th>
<th>Document 7</th>
<th>Modbus-E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrity authentication</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Authentication</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Functional code control</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Data transmission security</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

### TABLE II. EXPERIMENTAL TEST OF "WHITE LIST".

<table>
<thead>
<tr>
<th>Level</th>
<th>IP</th>
<th>Function Code</th>
<th>Device ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>192.168.1.100</td>
<td>all</td>
<td>all</td>
</tr>
<tr>
<td>50</td>
<td>192.168.1.50</td>
<td>1-6,F,10H</td>
<td>all</td>
</tr>
<tr>
<td>10</td>
<td>192.168.1.10</td>
<td>1-4</td>
<td>all</td>
</tr>
</tbody>
</table>

### TABLE III. TEST RESULT OF "WHITE LIST".

<table>
<thead>
<tr>
<th>PC</th>
<th>IP</th>
<th>Function Code</th>
<th>Device ID</th>
<th>Expected results</th>
<th>Actual results</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC1</td>
<td>192.168.1.100</td>
<td>3</td>
<td>10</td>
<td>legal</td>
<td>expected</td>
</tr>
<tr>
<td>PC3</td>
<td>192.168.1.10</td>
<td>6</td>
<td>5</td>
<td>illegal</td>
<td>expected</td>
</tr>
<tr>
<td>PC2</td>
<td>192.168.1.50</td>
<td>6</td>
<td>1</td>
<td>legal</td>
<td>expected</td>
</tr>
<tr>
<td>PC4</td>
<td>192.168.1.60</td>
<td>3</td>
<td>100</td>
<td>illegal</td>
<td>expected</td>
</tr>
<tr>
<td>PC4</td>
<td>192.168.1.60</td>
<td>6</td>
<td>14</td>
<td>illegal</td>
<td>expected</td>
</tr>
</tbody>
</table>
The server-side alarm log is shown in Figure 7.

**Modbus-E Efficiency Analysis**

The data of Figure 8 is obtained by the time difference between the response packet and the request packet captured by the packet capture tool in the middle of the transmission. This figure reflects the time it takes from sending a packet to receiving a packet. It can be seen from the figure that the time difference between the Modbus-E data packet and the original Modbus TCP data packet transmission is about 20ms, and the time difference is about 10ms compared with the literature 6. Compared with the security improvement, the difference can be ignored.

**ACKNOWLEDGMENTS**

This paper is supported by Jiangsu postgraduate research innovation project (SJCX18_0784).
REFERENCES