Research and Implementation of Modbus TCP Security Enhancement Protocol

Luo Xuan1,a, Li Yongzhong2

1School of Computer Science, Jiangsu University of Science and Technology, Zhenjiang Jiangsu 212003,China
2School of Computer Science, Jiangsu University of Science and Technology, Zhenjiang Jiangsu 212003,China

aXuan Luo: 851033149@qq.com

Abstract. Considering the security problem of Modbus TCP protocol, such as the lack of authentication mechanism, the protection mechanism of data transmission and abuse of function code, a secure industrial control communication protocol(Modbus-S protocol) is designed based on the original Modbus TCP protocol, which uses symmetric key algorithm to ensure the confidentiality of data; the uniqueness of data is guaranteed by the synchronization mechanism based on hash algorithm; digital signature algorithm is used to ensure the verifiability and integrity of data; finally, "White List" filtering mechanism is used to manage function codes based on roles to effectively prevent abuse of function codes. Through the verification and analysis of experiment, Modbus-S can fully compensate for the design defects of Modbus TCP protocol. Compared with the existing methods, this method has higher security and can comprehensively improve the communication security of Modbus TCP protocol.

1. Introduction
Industrial Control System (ICS) is widely used in petroleum, electric power, transportation, water conservancy and other Industrial enterprises that are the lifeblood of national economy. Information security of Industrial Control System is related to national security and social stability. With the continuous progress of integration of informatization and industrialization, industrial control system interacts with external network more and more closely. The traditional closed and isolated industrial control system is broken, and the security problem of external network is also introduced into industrial control system, which makes the network security problem faced by industrial control system more severe.

Modbus TCP protocol is the most commonly used communication protocol in industrial control system. In view of its security problems, many scholars have done relevant research. Gao Dongliang [1] designs a firewall of Modbus TCP protocol gateway based on netfilter. It filters the legitimacy of data based on "White List" filtering mechanism, but it can not solve IP forgery attacks and ensure the authenticity of data by relying on IP address filtering. Finally, the scheme can not prevent replay attacks. Steven [2] proposes a model-based intrusion detection method. The principle of this method is to establish an acceptable behavior model based on the protocol specification of the bus in the industrial control field, and detect potential attacks based on this model. Dayu Yang [3] proposes an intrusion detection technology based on anomaly detection. When the industrial control system is attacked, the related parameters of the system will fluctuate abnormally, such as CPU occupancy rate, equipment
parameters. The state matrix is established by comparing the network parameters under normal and abnormal conditions, and the alarm is triggered when the parameter matrix reaches the threshold. Both of the proposed schemes have extremely high requirements for accuracy setting. Once the accuracy setting is deviated, the detection effect will be seriously affected. Zhan Naisong [4] theoretically proposed the safety reinforcement method of industrial control system, but did not give specific design scheme from the technical level. Yang Jing [5] designed a security module based on trusted computing, which can realize data encryption and identity authentication, but the data processing time is too long to meet the real-time requirements. Fovino [6] guarantees data integrity, authenticity and replay attack by adding time stamps to the original protocol and signing the data. However, the data are still transmitted in plaintext, which can not prevent eavesdropping attacks. Secondly, there are still problems of abuse of function codes. Finally, this method determines time stamps based on NTP protocol, and its security is weak. If the NTP server is attacked or the attacker tampers with the information that the device interacts with the NTP server, the stability of the system will be destroyed. Hayes [7] proposes a hashing based stream transport control protocol (STCP), which supports source device authentication and has better integrity and authentication mechanism. However, the security of this scheme is based on the pre-shared key. Once the pre-shared key is cracked, the security will be destroyed.

In view of the problems in the above documents, an in-depth study on the abuse of function code, secure data transmission and feasibility of the scheme of Modbus TCP protocol is made in this paper. The Modbus protocol itself is improved and a Modbus security enhancement protocol (Modbus-S protocol) is proposed. In this protocol, replay attack can be resisted, both sides of sending data can be authenticated, key data fields are encrypt, and abuse of function codes can be prevented by "White List". Therefore, the security of industrial control network can be improved fundamentally and comprehensively.

2. Modbus TCP protocol

2.1. Introduction of Modbus TCP protocol
Modbus TCP protocol works in the application layer of TCP IP model and adopts master-slave communication mode. The master station can communicate with the slave station in one-to-one or many-to-many ways. Modbus TCP Application Data Unit (ADU) is composed of 7-byte protocol header (MBAP) and N-byte protocol data unit (PDU). Its message format is shown in Figure 1.

![Figure 1 Format of message of Modbus TCP](image)

2.2. Security analysis of Modbus TCP protocol
The security factors are not considered in the design of Modbus TCP protocol. The security risks of Modbus TCP protocol can be listed as follows:

(1) Lack of authentication;
Because there is no authentication mechanism between the client and the server, the attacker can pretend to send malicious instructions to both parties. For example, the "Stuxnet" virus that once shocked the world, the hacker is spoofing the virus, causing serious damage to the system.

(2) Abuse of function codes;
Since the protocol does not restrict the function code, any user who can access the client can execute any command. If the field staff may send a high-risk command to restart or shut down the device due to a misoperation, the server does not have a corresponding processing mechanism.

(3) Data tampering;
Modbus TCP protocol lacks the mechanism of verifying data integrity[8]. Attackers can intercept normal communication packets and tamper with them to attack the system.

(4) Data transmission in plain text;
The data packets transmitted by Modbus TCP are not encrypted, and some sensitive information is easily acquired by attackers. Attackers can collect relevant information to prepare for the attack of the system.

(5) Replay attack.
By observing the format of ADU of Modbus TCP, it can be seen that the transfer identification field is used to identify each transaction. However, replay attacks can not be prevented by this field. The attacker can send intercepted packets repeatedly, thereby destroying the stability of the system and making the system unable to work properly.

3. Design of Modbus-S protocol

3.1. Format of Modbus-S protocol
In order to realize authentication, tamper proof, anti-replay and function code abuse prevention, and at the same time to ensure the availability of the protocol requirements of the industrial control system, the Modbus TCP protocol is directly modified in this paper on the basis of no increasing the communication process, and a safe Modbus TCP protocol is proposed to achieve the data security of the industrial control system. The format of Modbus-S protocol is shown in Figure 2.

Figure 2 Format of Modbus-S protocol

Syn: Synchronized identification field;
Level: Privilege level field;
TLength: The length of the encrypted PDU;
En: The key of the symmetric key algorithm;
Pri: The key of digital signature;

3.2. Synchronization mechanism based on hash algorithm
Select a random number as the seed, calculate its hash value, and use the data of the first byte as the synchronization identification code. After each match is successful, the original hash value is hashed to compute the synchronization identifier for next time. And so on, update each synchronization identifier.
Using the unidirectional principle of hashing to ensure the uniqueness of the data, it can effectively resist replay attacks[9]. At the same time, there is no complete hash value in the data package, which greatly increases the difficulty of synchronization identification being cracked. The seeds of synchronous identification can be updated periodically, and the seeds of both sides can be synchronized by asymmetric key algorithm.

The synchronization principle based on hash chain is as follows:
(1) Generate random number as seed \(a_{i,1}\). When multiple clients communicate with the server, \(i\) is the number of their session. Calculate the hash value \(R_{i,1}\) of \(a_{i,1}\) and \(F(x)\) as a function to fetch data from the first byte of \(x\).

\[
R_{i,1} = hash(a_{i,1}) \quad (i=1,2,\ldots) \quad (1)
\]

\[
Syn_{i,1} = F(R_{i,1}) \quad (i=1,2,\ldots) \quad (2)
\]

(2) After each match is successful, calculate the next hash value and update it to \(R_{i,2}\). The first byte is taken as the synchronous identification \(Syn_{i,2} = F(R_{i,2})\).

\[
R_{i,2} = hash(R_{i,1}) \quad (i=1,2,\ldots) \quad (3)
\]

\[
Syn_{i,2} = F(R_{i,2}) \quad (i=1,2,\ldots) \quad (4)
\]

(3) Calculate the identification codes of each time by analogy.

Repeated use of hashing algorithm will increase the time overhead, but due to the short packet length of Modbus protocol, the increased time overhead is negligible.

3.3. "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 [10] settings are shown in Figure 3, the filtering process is shown in Figure 4.

| Level | IP Address | Device ID | Function Code |
|-------|------------|-----------|---------------|
| Level Check | legal | IP address Checking | legal |
| illegal | Device ID Check | illegal | |
| illegal | Function Code Check | illegal | Message discarded |
| Message passing |

Figure 3 The rule model of "White List"

![Diagram of "White List" filtering process]

Figure 4 The process of "White List" filtering

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 the values of coil and register, 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 whitelist based on role can effectively prevent the abuse of functional codes. The following is the local code of "White List" structure written by vB.net.
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 whitelist based on role can effectively prevent the abuse of functional codes.

4. Design and implementation of Modbus-S protocol

4.1. Design of Modbus-S protocol system

The conversion of Modbus TCP protocol to Modbus-S protocol is realized by a pair of communication software Modbus-S Client and Modbus-S Server. The principle of implementation is shown in Figure 6.

4.2. Processing of Modbus-S protocol data

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

(1) Modbus Client creates an effective Modbus request packet (M_req);

(2) The Modbus-S 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 (ILT), and PDU field encrypted with the key of Symmetric algorithm (En), construct N-Modbus field as shown in Figure 2, calculate its hash value by Hash algorithm, sign it with private key (Pr), It is then added to the N-Modbus and sent to the Modbus-S Server side.

\[ N-Modbus = Syn|Level|TID|PID|L|UID|ILT|PDU|En \]  \( \text{(5)} \)

\[ C = [N-\text{Modbus} | \text{Hash}(N-\text{Modbus})]Pr \]  \( \text{(6)} \)

(3) When Modbus-S Server receives the data packet, it determines firstly whether its synchronization identifier (Synsc) is identical with the synchronization identifier (Synss) 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.

\[ R = \text{Check(Synsc, Synss)} \& \text{Auth}(C) \& \text{WhiteListCheck}(C) \]  \( \text{(7)} \)
(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.

5. Experiment and analysis

5.1. Experimental environment
In this experiment, the Windows7 system virtual machine in VMware, Modbus Poll/Slave simulation software and Modbus-S Client/Server program developed by myself are used for simulation. The most commonly used AES symmetric key algorithm, RSA digital signature algorithm with 1024 bits key length and MD5 hash algorithm are used for analysis. The experimental environment is shown in figure 7:

![Figure 7 Experimental environment diagram](image)

PC1, PC2, PC3 and PC4 are all clients. Modbus Poll client simulation software and secure communication software Modbus-S Client are all installed on them. 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-S server side communication software. Modbus-S protocol is used to communicate between PC1 to PC5 and original Modbus TCP is used to communicate between PC5 and security gateway.

5.2. Theoretical analysis of security of Modbus-S
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.

| Protocol security analysis | Integrity authentication | Authentication control | Functional code control | Transmission security |
|----------------------------|--------------------------|------------------------|-------------------------|----------------------|
| Document5                  | yes                      | yes                    | no                      | yes                  |
| Document6                  | yes                      | yes                    | no                      | no                   |
| Document7                  | yes                      | yes                    | no                      | yes                  |
5.3. Test of replay attacks
The experiment used TCP/UDP debugging tool to replay intercepted data packets. Figure 8 shows the normally communicating packets grab by Wireshark.

Figure 8: Packets of normal communication

Figure 9 shows the setting interface of playback data.

Figure 9: Setting interface of playback data

Figure 10 shows the results of the replay attack.

Figure 10: Results of replay attack
5.4. Test of eavesdropping attack

Wireshark will be used to capture the packets of Modbus-S protocol during normal communication for analysis. Figure 12 shows ADU data of modbus-S request packet. Figure 13 shows ADU data of modbus-S response packet. As can be seen from the two figures, the key fields of the data have been encrypted, the test results are consistent with expectations.

5.5. Filtering test of "White List"

Table 2 is a "White List" of the experimental tests, Table 3 is the test results, Figure 14 is a data communication packet and the server side alarm log is shown in Figure 15. The test results 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.
| PC     | IP           | Function Code | Device ID | Expected result | Actual results |
|--------|--------------|---------------|-----------|-----------------|----------------|
| PC1    | 192.168.1.100| 6             | 1         | legal           | legal          |
| PC3    | 192.168.1.10 | 6             | 5         | illegal         | illegal        |
| PC2    | 192.168.1.50 | 6             | 3         | legal           | expected       |
| PC4    | 192.168.1.60 | 3             | 7         | illegal         | expected       |
| PC4    | 192.168.1.60 | 6             | 4         | illegal         | expected       |

5.6. Analysis of performance of Modbus-S

The data of Figure 16 is obtained by the time difference between the response packet and the request packet captured by the packet capture tool during transmission. The figure reflects the time taken from sending the request packet to receiving the response packet. It can be seen from the figure that the time difference between Modbus-S and original Modbus TCP data packets is about 8 ms, and that between Modbus-S and document 6 is about 2 ms. Compared with the security improvement, the difference can be ignored.
6. Conclusion

In this paper, the Modbus-S protocol is designed by analyzing Modbus TCP protocol and studying its inherent security defects. By running the corresponding software and processing the data flowing through the industrial control field, finally realize the safety reinforcement of the industrial control system.

The communication software Modbus-S Client can be deployed directly on the original client computer terminal. However, since the original server is usually a security gateway, it is not convenient to deploy the Modbus-S server directly. It is necessary to deploy the computer terminal to process data before the security gateway. The next step we will study and design the corresponding hardware modules to reduce implementation costs.

Acknowledgment

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

References

[1] Gao Dongliang. Research And Implementation Of Modbus TCP/IP Protocol Firewall[D]. Beijing University of Posts and Telecommunications, 2015.
[2] Steven Cheung, Bruno Duterre, Martin Fong, et al. Using model-based intrusion detection for SCADA networks[C]// Proceedings of the SCADA security scientific symposium. 2007, 46:1-12.
[3] Dayu Yang, Alexander Usynin, J. Wesley Hines. Anomaly-based intrusion detection for SCADA systems[C]// 5th intl. topical meeting on nuclear plant instrumentation, control and human machine interface technologies (npic&hmit 05). 2006:12-16.
[4] Zhan Naisong, Qiao Zhenya. The Research on Information Security Protection of Industrial Control System[J]. Cyberspace Security, 2017, 8(12): 66-70.
[5] Yang Jing. Research on Modbus/TCP Protocol Security In SCADA Systems[D]. Beijing University of Technology, 2016.
[6] Fovino I N, Carcano A, Masera M, et al. Design and implementation of a secure modbus protocol[C]// International conference on critical infrastructure protection. Springer, Berlin, Heidelberg, 2009: 83-96.
[7] Hayes G, El-Khatib K. Securing modbus transactions using hash-based message authentication codes and stream transmission control protocol[C]// Communications and Information Technology (ICICT), 2013 Third International Conference on IEEE, 2013: 179-184.
[8] Xiao Liang, Li Qiangda, Liu Jinliang. Survey on Secure Cloud Storage[J]. Journal of Data Acquisition and Processing, 2016, 31(3): 464-472.
[9] Liu Fei, Zhang Renbin, Li Gang, etc. Security authentication protocol of Modbus/TCP based on hash chain and synchronization mechanism[J]. Application Research of Computers, 2018, 35(4): 1169-1173.
[10] Zhang Renbin, Li Sixian, Liu Fei, etc. Research on fine-grained filtering algorithm based on Modbus function code [J]. Application Research of Computers, 2018, 35(1): 277-281.