Modware: Securing the Modbus Protocol

Mohammad Eshan[1] and Chris Tremblay[2]
{me3031[1],cst1465[2]}@rit.edu

March 2023

1 Project Idea

The Modbus protocol is widely used in Industrial Control Systems (ICS) to exchange data between devices and controllers. However, Modbus was not designed with security in mind, leaving devices vulnerable to MITM, spoofing, and other forms of attacks. The Modbus Organization has proposed a solution called `Secure Modbus` that is specified with TLS v1.1 and v1.2 but, not the newer and more secure v1.3. Even with the development of `Secure Modbus` many sensors and industrial equipment do not even support TLS within their network stack and are slow to adopt such technologies until they are proven reliable, efficient, and worth the effort to integrate. This renders `Secure Modbus` infeasible in most cases. Our project proposes to make an unobtrusive client/server middleware that encrypts communication without having to change existing hardware or spend lots of time integrating. Since the traffic from the devices is not secure, the middleware is intended to reside very close (same switch, or even in-line) to the device to reduce the amount of time traffic is unencrypted.

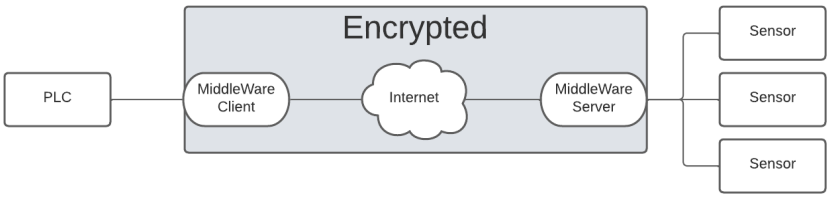

Figure 1: Basic Architecture

2 Literature Review

Modbus security is not abundant in academic literature. One reoccurring solution observed was to add TLS into the communication stack. The Modbus Organization themselves provided the protocol specification [1], and researchers from China also provided a similar solution [2]. Adding TLS to the communication stack is very viable however, legacy and lower-end equipment may not have a sophisticated enough communication stack to implement this solution. Nor would legacy equipment have the resources to do their designed task and run intense cryptographic computations.

The authors in [3] provide a comprehensive solution to Modbus that supports roles, privilege levels, and non-repudiation of human uses. However, it does require a server running middleware between the communicating devices. This MITM facilitates all of the security functions between the devices. This was done on top of a software package called Node-RED, which provides a subset of security features already. This is a very sophisticated and feature-dense solution, which is desirable for companies that are up-to-date on modern computing and security concepts. This could be less desirable for companies who see this as feature bloat and do not need to utilize that whole security suite.

The authors in [4] propose a system that slightly modifies the Modbus protocol and calls for a MITM server. They propose the addition of a digest appended at the end of the Modbus packet, along with signing the whole packet with an RSA private key. They also suggest adding a timestamp within the digest. All 3 of these features give integrity, non-repudiation, and freshness against replay attacks. However similar to [3], they do require a server running middleware to facilitate the communication between end devices.

3 Timeline

Week 9	Spring Break / Configure Development Environment
Week 10	Design Software and Begin Basic Development
Week 11	Development of Client/Server Middleware
Week 12	Development of Client/Server Middleware
Week 13	Deploy Solution in Simulated Environment and Benchmark
Week 14	Buffer Week and Report Writing

References

[1] I. Modbus Organization, "Mb-tcp-security-v21_2018-07-24.pdf," Jul 2018.

[2] W. Jingran, L. Mingzhe, X. Aidong, H. Bo, H. Xiaojia, and Z. Xiufang, "Research and implementation of secure industrial communication protocols," in *2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS)*, p. 314–317, Mar 2020.

[3] T. Martins and S. V. G. Oliveira, "Enhanced modbus/tcp security protocol: Authentication and authorization functions supported," *Sensors*, vol. 22, p. 8024, Oct 2022.

[4] I. N. Fovino, A. Carcano, M. Masera, and A. Trombetta, "Design and implementation of a secure modbus protocol," in *Critical Infrastructure Protection III* (C. Palmer and S. Shenoi, eds.), IFIP Advances in Information and Communication Technology, (Berlin, Heidelberg), p. 83–96, Springer, 2009.