Experimenting Security Algorithms for the IEC 61850-based Substation Communication

Sugwon Hong[1], Dae-Yong Shin[1], and Seung-Jae Lee[2]
[1]Department of Computer Software, [2]Electrical Engineering
Myongji University

ABSTRACT: *Security becomes the major concern in a modernized electricity network. The IEC 61850 standard has been developed for substation automation, providing more enhancing communication functionalities. The standard defines several kinds of messages for data exchange between nodes in the substation. Among them, two critical messages have very stringent performance requirement for secure substation operation and protection. In this paper we design a security protocol for guaranteeing message authentication and integrity. In order to show whether the protocol can achieve the performance criteria imposed by the standard, we conduct some experiments on a desktop PC and an embedded device.*

1. Introduction

Major concern about cyber attack in the power grid stems from the notion that the electric network is no longer an isolated network which prohibits outsiders from entering the network, nor is the specialized network based on private platforms and protocols, allowing only technical staffs with special knowledge to access to the resources. The communication architecture is more relying on the open standard communication protocols. The use of the open communication protocols renders the control or protection devices more vulnerable to cyber attacks in many applications [1].

IEC 61850 is the standard for the design of electrical substation automation and for provision of more enhanced communication functionalities as well. The IEC 61850 communication protocol defines the ways of exchanging messages between nodes in the substation [2]. Among the several types of messages defined in the standard, the sample value message and the General Object Oriented Substation Event (GOOSE) message are critical for secure operation of protection devices in the substation. Because of their importance in the secure substation operation, any compromising of these messages should be prohibited. Along with the requirement of secure transmission, these messages have very stringent performance requirement since they have to be processed in real time.

We consider the message authentication and integrity as the security goal for secure substation operation, since false sample value messages or malicious GOOSE messages may cause disastrous result in the substation protection

This work was supported by the ERC program of MOST/KOSEF (Next-generation Power Technology Center).

operation. The security objectives are to assure only authorized access within the system, and prevent spoofing and playback of captured data from non-trusted entities.

As important as the security protocol is, the implementation of the security algorithms in real-world environment is also critical since most devices in the substation are embodied in embedded systems, thus having very limited computing powers and memories to process the cryptographic logic. In this paper we do some experiments to evaluate the feasibility of the proposed security protocol on a desktop PC and an embedded device.

2. IEC 61850-based substation communication

Main components of the substation are intelligent electronic devices (IED), power equipments, and substation controller. The substation controller, located in a central site, monitors and supervises a large number of IEDs which are field devices located in physical environments. IEDs gather data from sensors which measure current and voltage, and send data to the substation controller. The actuator as a part of IED controls the operation of power equipments by commands issued by other IEDs. The substation controllers have a hierarchical structure. A high-level master station can control several sub-master stations.

The data and command transfer takes place between the substation controller and IEDs, between IEDs, or between IED and sensors (or switchgears). The transferred information is carried over the substation network. The substation network is based on various communication channels and network technologies including Ethernet, serial links, wireless communication, and so on. The communication between the substation devices is governed by the standard communication protocols. The most commonly used protocols are IEC 60870-5, DNP3 and Modbus. Recently the International Electrotechnical Commission (IEC) is working on the new protocol, IEC 61850, which not only defines a new structure for substation automation, but also can provide more enhanced communication functionalities [2]. The IEC 61850 defines seven different types of messages which are exchanging between the substation nodes. The communication stacks depending on the message types are shown in figure 2.

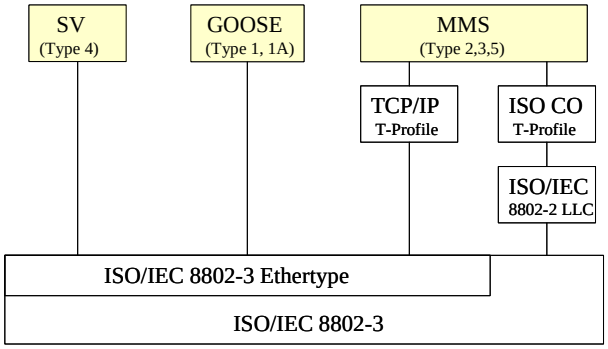

Figure 1. The protocol stacks for the IEC 61850 messages

Among the messages, the message of sampled values is intended to deliver samples of 960 Hz signal from measuring devices to IEDs. The General Object Oriented Substation Event (GOOSE) message is an urgent message which conveys protection information between IEDs and circuit breakers or other protection devices. Upon detecting an event, the IED multicasts GOOSE messages to notify other IEDs of the events, and cause an actuator to do protection action. The GOOSE message transmission has stringent performance requirement. No more than 3 ms is allowed to elapse from the time an event occurs to the time a message is transmitted. Collision is possible since the IEC 61850 is based on IEEE 802.3 network. So the GOOSE messages are retransmitted several times by IED.

The Manufacturing Message Specification (MMS) message is intended to configure and supervise the different devices in the substation. For this reason, unlike the other two messages, the MMS message has low or medium speed, consequently loose performance requirement.

There are two possible communication modes. The first one is the controller-IED communication mode where data transfer is done over a path between the substation controller and IEDs. The other one is the peer-to-peer mode where data can be delivered between IEDs directly. The current communication protocols only support the controller-IED communication mode, while the IEC 61850 protocol support the peer-to-peer mode too. In this paper we will consider the security protocols which can be applied to both modes.

3. Security Protocol

3.1 Goal

In this paper we focus on secure transfer of the sample value message and the GOOSE message. The secure transmission of these two messages plays a critical role for normal protection operation in the substation. Receivers need to verify that messages are sent from claimed senders. Attackers can inject malicious messages to the IEDs or breakers, consequently causing system malfunctions. To authenticate the owner of messages is one of the most important security requirements in many applications in the supervisory control and data acquisition (SCADA) system.

Receivers also need to make sure that the messages they receive are not altered on the way by attackers. In particular, sample values are used for the IEDs to decide whether voltage, current, or frequency anomalies happen. If these values are modified, the IEDs are mistaken to understand the current status. For this reason we consider the security protocol to guarantee the message authenticity and integrity.

3.2 Message Authentication and Integrity

In this paper we are using the message authentication code (MAC) to verify the authenticity of the sender and the integrity of the message. Two methods are used

for computing MAC. One is just to use a block cipher. The procedure is simply to encrypt a message in cipher block chaining (CBC) mode and discard all cipher text block except the last block [3]. The last encrypted block is used for MAC. The block cipher algorithms we use here are Advanced Encryption Standard (AES) and SEED [4,5].

The other method is to use the Keyed-Hashing (HMAC) [6]. In the HMAC, first, the sender A concatenates the Sync Code C, the original message M_A, and the (session) authentication key, K_{AB}, then computes MAC by applying a one-way hash function, H, to the concatenated message. The detailed procedures are explained in [6]. Next, the sender replaces the authentication key by the MAC and finally delivers the message.

The node B applies the same hash function to obtain new MAC on the message it received with the authentication key. If these two MACs are the same, B can trust that the message was sent by the claimed sender A, and also the message was not modified on the way.

The authentication key can be of any length. But it is recommended that the key length should not be less than L bytes which is the byte-length of the hash function output, since it would decrease the security strength. Keys longer than L bytes are acceptable but the extra length would not provide significant increase of security [6]. In the implementation we use L=16 as a default secrete key length since the default keyed hash function is MD5 [7].

The Sync Code is used to verify the freshness of the message. Since the Sync Code is a non-decreasing number, the value of a new message should be bigger than the one of an old message. Comparing these two values reveals whether the message was resent or not, thus ensuring that no attackers replay old messages.

3.3 Key Distribution

Because the substation controller or server is the base node which communicates with the other nodes in the network, compromising the server will cause the whole network to be out of service. Generally the server is deployed in the protected location. We assume that the server is a trusted base, and all IEDs trust the server at the initial setup. At the creating time the server is given two master keys. One key is the master encryption key used for encrypting the session key which is the authentication key that two communicating parties share. The other is the master authentication key used when the server delivers the session authentication key to IED. Each IED is also given two same master keys which are shared with the server at the creation time, and the session authentication keys are distributed from the server whenever necessary. And we assume that each node keeps the master keys without any harm.

For the key distribution between the server and IEDs, we apply the same simple and secure algorithm proposed in [8]. So here we focus only on the peer-to-peer

communication mode. Direct message exchange between IEDs or between IEDs and power apparatuses will be possible if the IEC 61850 standard is prevalent as the standard communication protocol.

Many protocols have been proposed for the three party server-based key distribution protocol from the Needham-Schroeder protocol to the ISO/IEC 11770-2 server-based protocols and the Kerberos protocol which are widely accepted as the de facto standard protocol in the network community [9, 10]. Even though the protocol proposed here has some similarities with the ISO/IEC 11770-2 and the Kerberos protocol, it has unique characteristics, reflecting the constraints of the system components in the SCADA system.

First, Any IED can initiate the key request to the server as the same way carried in the server-to-IED communication mode. This provides consistent operation to the server to process the key request regardless of the communication modes. According to the address fields in the key request, the server can differentiate which operation it should take.

Second, every difficult and delicate job is given to the server's shoulder, including random number generation and key generation. The case the IED has to generate a nonce is only the first step. Other than that, IEDs can get away with the obligation of clock synchronization or random value generation which is the delicate parts of doing security operation. At every message exchange, message authenticity is guaranteed from MAC using the master authentication keys. The final message exchange provides the key confirmation to both nodes.

4. Experimental Result

4.1 Configuration and Result of desktop PC

The test configuration consists of four PCs each of which simulates a merging unit (MU) which generates the sample values, a substation controller, a protection IED, and an actuator IED which sends a trip signal to a circuit breaker (CB) as shown in figure 2. All devices are operating on the 1 gigabit Ethernet. The MU sends sample values to the IED with 960Hz speed. If the protection IED senses any anomalies in the sample values, it notifies the actuator of the event by sending the GOOSE messages. Then the actuator opens or closes the CB based on the information of the GOOSE message.

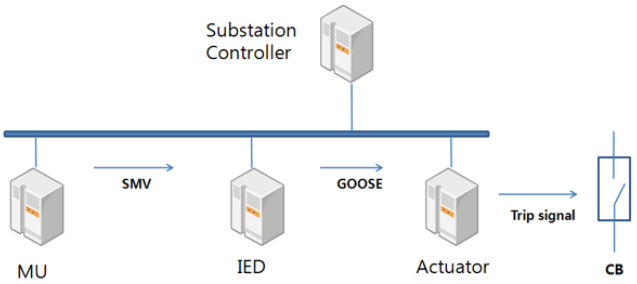

Figure 2. The experimental system configuration

The security protocol for the message authentication and integrity is applied to every sample value and GOOSE message to be transmitted at the MU and the IED. The message length of sample values in the experiment is 128 byte fixed size. Table 1 and 2 show the specifications of PC and software used in this experiment.

Table 1. Specification of desktop PC

Parameters	Specification
Processor	Intel Pentium 3
OS	Linux 2.6.18-128
Clock Speed	450Mhz
RAM	SDRAM 256Mbyte
Hard Disk	Seagate 80G(7200RPM)
Ethernet Controler	Realtek 8139D

Table 2. Specification of Cryptographic software

Algorithms	Library	Optimazation
AES,SEED	openssl-0.9.8k	Not optimized
MD5, HMAC-MD5	openssl-0.9.8k	Not optimized

In this experiment we measure the time which it takes to execute the MAC algorithms for the message authentication and integrity. The result is shown in table 3. The values are the average of 70,000 measurements, discarding the first 100,000 measurements. The table shows two execution times which were obtained from two platforms, the desktop PC and the embedded system(ARM).

Table 3. Comparison of crypto computation time

Encryption	MAC	time(us)	
		ARM	PC
AES256		2.25	15.05
AES256	SEED	5.89	38.03
AES256	MD5	3.68	21.31
AES256	HMAC	5.51	38.96
SEED		3.4	19.23
SEED	AES256	5.81	37.97
SEED	MD5	4.86	29.56
SEED	HMAC	6.85	49.1
	HMAC	3.68	23.63

As explained in section 3, we implement two methods for computing MAC. One is just to use a CBC of the block ciphers. The other method is to use the keyed-hashing (HMAC). In this table the first column shows the cryptographic algorithms used for encryption, and the second column denotes the algorithms used for computing MAC. The AES256 in the first row which denotes the key length of 256 bits means using the first method for computing MAC, since the MAC is

automatically computed as the last encrypted block by the CBC mode. And the HMAC in the last row denotes that MD5 is used for hash algorithm in the keyed hashing without encryption. Even though we are interested in two MAC algorithms, we show the results of several combinations of cryptographic algorithms to verify that the computation has been done correctly.

Based on the result, we can derive the following observation. First the time to be required for processing the message authentication and integrity algorithm is reasonably small, satisfying the stringent timing requirement of the GOOSE message transmission. Also the security algorithms do not pose any obstacle to handle the regular spate of sample value messages with 960Hz speed.

Second, the keyed-hashing (HMAC) method gives the better performance than the block cipher MAC, since HMAC can avoid any encryption procedure. But the improvement is not significant because the lengths of the sample values and GOOSE messages are extremely small compared to normal messages.

Finally, as expected, when we do the message encryption as well as the MAC computation, we encounter the significant increase in the message processing time. Therefore we need to avoid the message encryption unless the message confidentiality is required.

4.2 Result of embedded device

The same computation has been done on an embedded system. Table 4 shows the specifications of the embedded system and software used in this experiment.

Table 4. Specification of embedded device

Parameters	Specification
Processor	Intex Xscale PXA270
OS	Linux 2.6.12
Clock Speed	520MHz
RAM	SDRAM 128MByte (256Mbit(32Mbyte) *4EA)
Flash Memory	Intel Strata Flash 64Mbyte
Ethernet	10/100M LAN91C111 Ethernet Controller
Instruction Cache	32KB
Data Cache	32KB

In this experiment, we focus on how the system can receive and process incoming packets when packets are arriving from multiple sources. In order to see how many incoming packets from MUs the processor can handle, we count the number of packets stored in the socket buffer (received packets) and the number of packets to be processed for cryptographic computation depending on the number of packets arrived. Figure 3 shows the result. As shown in the figure, in the desktop PC, all packets sent by MUs are stored in the socket buffer, and cryptographic algorithms for all incoming packets in the buffer are computed by a process until there are approximately 40 MUs, which mean that approximately 40,000 packets are

generated. Beyond this number, packet loss happens in the socket buffer. At the same time the number of processed packets begins to skew from the number of packets arrived at the buffer. On the contrary, in the embedded system, packet loss happens even when packets arrive from the very small number of sources.

This deviation can be explained by the way CPU is doing operations inside the system. Figure 4 shows the internal operations which take place inside the embedded device to handle the incoming packets. As shown in this figure, there are two main operations of the processor. The first one is the polling of packets in the socket buffer by which the packets are read into the shared memory. The second part is to do cryptographic computation. The processor takes turn on doing these two tasks. The experiment shows that the processor reads the packets from the buffer or computes the cryptographic logic in the bulky way when the number of packets exceeds the limit. For example, when the CPU reads the bulky number of packets, the computation is delayed and consequently more packets should wait for processing in the shared memory.

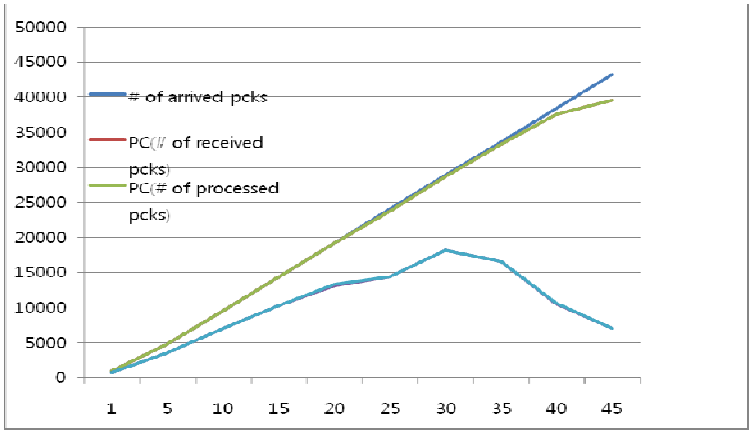

Figure 3. throughput vs. # of sources

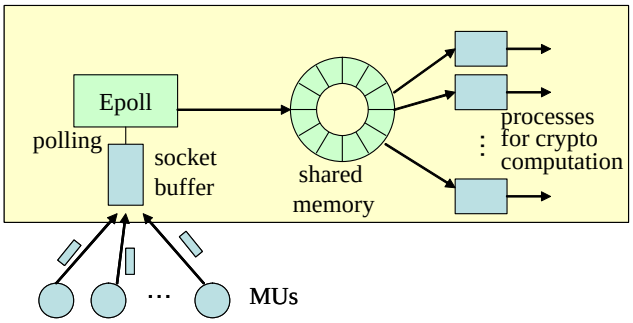

Figure 4. Operation to process packets in the embedded device

5. Conclusion

In this paper we consider the security protocol to provide the message authentication and integrity for transmitting the sample value message and GOOSE

message defined in the IEC 61850 standard. We examine whether the security algorithm can meet the performance criteria specified in the standard by doing experiments. We show that the HMAC algorithm give the better performance than other message authentication algorithm even considering small size of the sample value or GOOSE messages. On the embedded system which has limited computing power and memory, the experiment shows that the packet loss can happen, and at the same time not all arriving packets can be processed. To analyze the results we need precise understanding of the processor and its internal operation. And to improve the packet processing in the embedded system, to do parallel computation on the multi-core platform can be one solution. All this leaves further study.

References

[1] J. Slay and M. Miller, “Lessons learned from the Maroochy Water Breach,” IFIP Springer Boston, vol. 253, pp73-82, 2007.
[2] IEC, “Communication networks and systems in substation ,” IEC 61850-8-1, 2004.
[3] M. Stamp, Information Security: Principles and Practice. New York: Wiley, 2006, p. 55.
[4] J. Daemen and V. Rijmen, The Design of Rijndael: AES - The Advanced Encryption Standard, Springer-Verlag, 2002.
[5] H.J.Lee, S.J.Lee, J.H. Yoon, D.H.Cheon, and J.I.Lee, “The SEED Encryption Algorithm,” IETF rfc 4269, December 2005.
[6] H. Krawcryk, M. Bellare, and R. Canetti, “HMAC: Keyed-Hashing for Message Authentication,” RFC 2104, IETF, February 1997.
[7] F. Baker and R. Atkinson, “RIP-2 MD5 Authentication,” RFC 2082, IETF, January 1997.
[8] I.H.Lim, S. Hong, M.S.Choi, S.J. Lee, S.W.Lee, B.N.Ha, "Applying Security Algorithms against Cyber Attacks in the Distribution Automation System," IEEE PES, April 2008.
[9] B. Clifford Neuman and T. Tso, “Kerboros: An authentication service for computer networks”, IEEE Communications Magazine, Vol. 32, No. 9, pp33-28, September 1994.
[10] ISO, Information Technologies – Security Techniques –Key Management – Part 2: Mechanisms Using Symmetric Technique ISO/IEC 11770-2, 1996, International Standard.