Research on A-Key Distribution Algorithms For Protecting Data of RS-485-based Industrial Infrastructure

Jae-gu Song, Sungmo Jung, Seoksoo Kim
Department of Multimedia
Hannam University
Ojeong-dong, Daedeok-gu, Daejeon 306-791
KOREA
bhas9@paran.com, sungmoj@gmail.com, sskim0123@naver.com http://www.maplabkorea.com

Abstract: - The RS-485 protocol is a Modbus serial communication, mainly used by SCADA(Supervisory Control And Data Acquisition). Most industrial infrastructure using this communication protocol is designed not to be exposed to an open network environment, seriously vulnerable to security threats. Therefore, this study examines key management architecture in order to suggest a key exchange algorithm suitable for the RS-485 protocol. The suggested algorithm can support both 1:1 and 1:N communications as well as minimize the amount of loads arising from encoding/decoding so as to increase its application.

Key-Words: - RS-485, Modbus, SCADA, Security, Infrastructure, Key Distribution Algorithms, Key Management

1 Introduction

More recently, security for industrial infrastructure is of growing interest. This is because of the current shift in terrorist attacks from physical assaults from cyber attacks targeting the major infrastructure of a society. Most industrial infrastructure involves electricity, water systems, nuclear power, and gas distribution networks, of which even a minor problem can result in tremendous impacts on a society.

There are at present many studies on protection of such social infrastructure. In particular, a greater possibility of utilizing the Internet by industrial networks calls for an effective measure to overcome security weakness among internal communications protocols[1, 2, 3]. Previous communications protocols of the infrastructure are closed, and communications are done based on reliance between physical devices. That is, the attempt to connect them with the Internet may present a significant vulnerability to threats existing in a TCP/IP environment. Therefore, the protective measure shall be focused on communications protocols used by current industrial infrastructure.

Of the communications protocols used by the industrial infrastructure, this study is focused on Modbus, and suggests an authentication key distribution algorithm of encryption for safe data transmission.

Modbus is one of the major communication means which has been mostly used by production facilities worldwide since 1979. Due to the simple protocol structure and various applications, Modbus has been employed by most national infrastructure [4] [5]. As a result, weakness of Modbus is considered the same as that of the national infrastructure [6]. Thus, this study analyzes the vulnerability of Modbus protocols, especially, RS-485, and suggests A-key distribution algorithms for safe data transmission.

2 Related Research

2.1 Modbus serial communications

Modbus communications are largely divided into three types, Modbus serial, Modbus plus, and Modbus TCP/IP [7]. Of these, Modbus serial is again classified into one based on RS232C, RS-422, and RS485. According to transmission types, it could be divided into RTU (Remote Terminal Unit) and ASCII (American Standard Code For Information Interchange)[8].

Communications of Modus Serial is done by token passing, in which Master can have only one connection at a time and Client, can have 247 connections. The request mode of Master supports both Unicast and Broadcast. In the Unicast mode, Client replies a request of Master and 1~247 of addresses are available. In the Broadcast mode, Master sends a request to all Clients and 248~256 of

addresses are available. The speed is 1200/2400/4800/19200bps, 56Kbps and 115.2Kbps while the maximum communications distance is about 1 km and termination is 150Ω(Ohms)/0.5W.

A general communication frame is composed of a header, data, and a trailer[9, 10].

In order to generate an authentication key, we need to understand characteristics of data, frame elements, and a function code.

Modbus has the following frame elements.

Primary Tables	Object Type	Type of Access
Discrete Input	Single Bit	Read-Only
Coils	Single Bit	Read-Write
Input Registers	16-bit word	Read-Only
Holding REgisters	16-bit word	Read-Write

Table 1. Frame Elements of Modbus

In this study, data encryption keys are generated by understanding Modbus function codes.

Table 2 shows the Modbus function codes.

2.2 Encryption System

Modern day cryptology involves encryption systems, encryption analysis, authentication, e-signature, and so on.

A message to be protected by cryptology is called a plain text and the one converted by a cytological method, a cipher text. In this case, the process of changing a plain text into a cipher text is encryption while the reverse process is description[11].

The purpose of cytological services is as follows.

①To prevent inadequate exposure. Unauthorized users have no access to a text.

②To prevent inadequate modification. Unauthorized users have no right to change a text.

③To prevent inadequate denial of service

④Not to allow one who has sent or received a text to deny such a fact

2.3 A-Key and encryption methods

We need to understand key management methods for key distribution. Major key management methods are shown in Table 2.

Public Function Code Definition				Function Codes	hex
Data Access	Bit access	Physical Discrete inputs	read discrete inputs	02	02
		Internal bits or physical coils	read coils	01	01
			write single coil	05	05
			write multiple coils	15	0F
	16 bits access	Physical Input Registers	read input register	04	04
		Internal Registers or Physical output registers	read holding registers	03	03
			write single register	06	06
			write multiple registers	16	10
			read/write multiple registers	23	17
			mask write register	22	16
			read FIFO queue	24	18
	File record access		read file record	20	14
			write file record	21	15
Diagnostics			read exception status	07	07
			diagnostic	08	
			get com event counter	11	0B
			get com event log	12	0C
			report slave ID	17	11
			read device identification	43	2B
Other			encapsulated interface transport	43	2B

Table 2. Modbus Function Code

2.3.1 Centralized Group Key Management

One of the basic methods is that the central server shares a secret key with other members and delivers a group key to each member through encryption using the shared secret key[12].

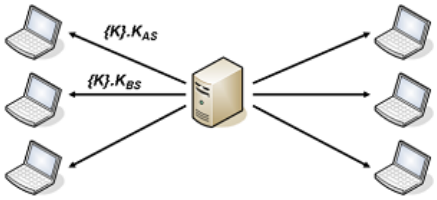

Fig.1 Centralized Group Key Management Protocols

2.3.2 Logical Key Hierarchy

In view of the determinacy of the centralized group key management, logical key hierarchy(LKH) has been suggested with the following features[13].

▪Each node maintains one key.
▪The root node of a tree becomes a group key.
▪Each terminal or user is connected with one node and should receive all keys of that node and its ancestor node.

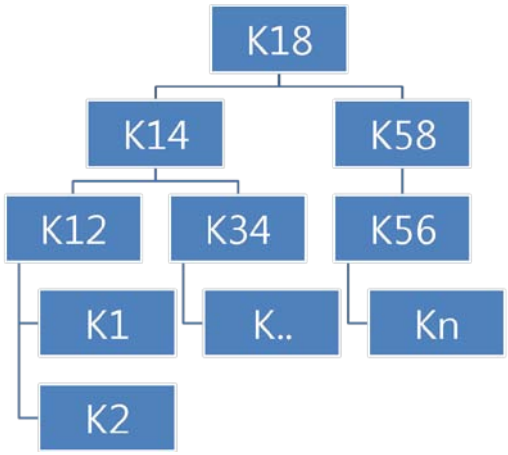

Fig.2 Logical Key Hierarchy

2.3.3 Comparison of block encryption algorithms

A block encryption algorithm divides a plain text according to the given length and carries out encryption based on the block unit. In transmitting/receiving data in the RS-485 protocol, encryption time and data size must be considered. Some of the major block encryption algorithms are AES, SEED, and ARIA. The following Table 3 compares them [14] [15].

(Unit : cycle/byte)

CPU	ARIA	AES	SEED
Pentium 3	37.3	23.3	42.4
Pentium 4	49.0	30.5	81.3

Table 3. Comparison of functions of block encryption algorithms

Although AES may have the best function, this study uses SEED, which is considered as an international standard algorithm.

3 Requirements for Authentication Key Management

In order for A-key management, one must consider features of a private key and a public key. The biggest weakness of a private key encryption system is difficulty of key management, for an administrator shall manage a huge number of keys. A public key encryption system overcomes such weakness. In the system each user receives two keys, a public key and a private key [16]. Then, each user only needs to manage his own private key. A public key encryption system requires a public key management system (public key directory) to manage users' public keys and each user has free access to the directory [17].

This study has derived requirements for key management by analyzing KDC and GKMP, two of authentication key distribution methods.

As for KDC, one KDC exists and manages keys for members of all groups, which renders a simple structure as shown in figure 3. It is more effective when the groups are small but quite ineffective if the groups are large or the members frequently change. Such structure is not suitable for a sensor network with limited resources as in this study. Therefore, there should be a measure to deal with frequent changes of members by applying a single key management program.

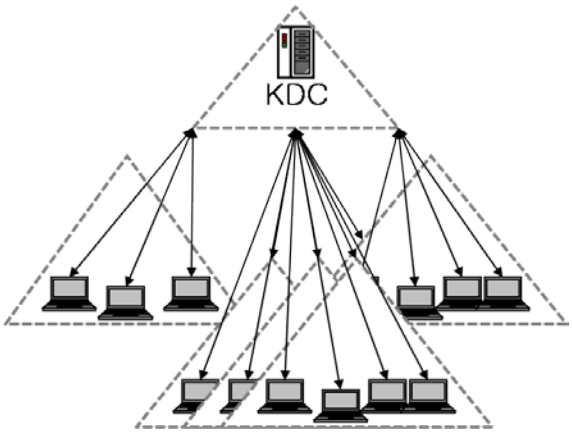

Fig.3 KDC Algorithm

When the number of groups increases, one KDC shall manage a multitude of groups, which can lead to a sudden rise in a communication load [18]. With a single key, it could be difficult to distribute keys to members, which frequently change.

GKMP generates a group key before creating groups. The group key is made by a group administrator and a random member of a group [19, 20]. Here, they make two keys, Group Traffic Encryption Key(GTEK) and Group Key Encryption Key(GKEK).

Group Key Packet(GKP) = [GTEKn, GKEKn+1]

After creating GKP, a group controller contacts each member, authenticates them, and encodes GKP into SKEK (Session Key Encryption Key) in order to send it to members. In case of renewing a group key, the group administrator randomly selects a member and makes new GKP with him. The new GKP is encoded by GKEK of the previous GKP and broadcasted to the group [21]. When a member sends a request to join the group, the group administrator authenticates a new member, creates SKEK for the member, and distributes GKP in a similar manner. There could be two cases that a member should be removed. First, a member is removed by mutual agreement. Here, a target member receives a removal message from the group administrator and he agrees to the removal by sending a message also. Second, when a member violates security rules, he is removed by force. Here, remaining members change their group key.

It is advantageous that each group controller is in charge of only a small number of group members and bottleneck phenomenon is less expected compared with use of one KDC. However, GKMP is managed through a central control center also, which means key issuance and update process becomes complex due to frequent key generation. As a result, it requires direct authentication between groups or members as shown in figure 4.

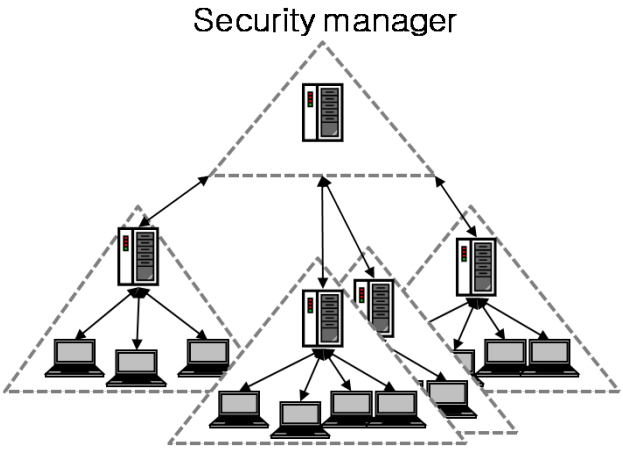

Fig.4 GKMP Algorithm

One of the features of the industrial infrastructure is that RTU may frequently join or leave, which requires a solution. To that end, the concept of ECSM(extended Complete Subtree Method) is applied. ECSM creates a pre-defined key tree by N(the biggest number of group members) and distributes a new key using a subtree root key in order to give an A-key to final nodes that frequently join or leave a group as shown in figure 5[22].

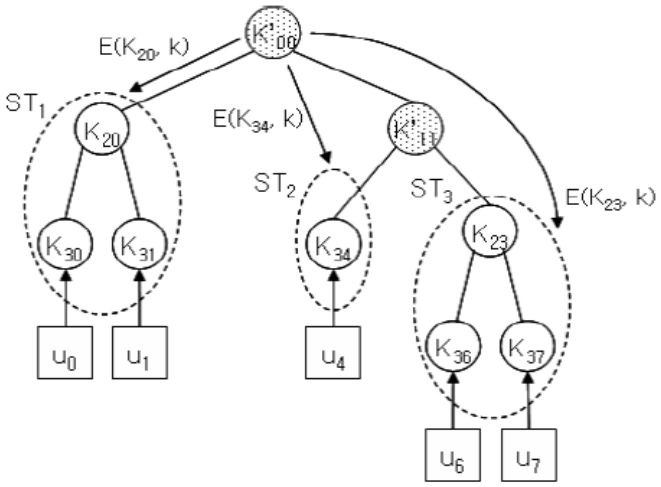

Fig.5 ECSM Algorithm

Finally, there should be a method of renewing a batch key for a broadcast environment as shown in figure 6. For this, research needs to be done on management of a key before the rekeying and an additional key. In general, a batch key shall manage the number of group members, tress, subtrees, length of a key tree, a node key, depth of a tree, and previous keys before rekeying.

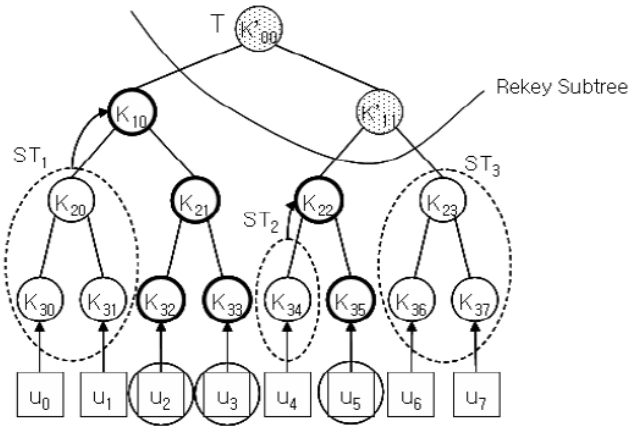

Fig.6 Management of a batch key

N = Number of group members
T = Tree
STK = Subtree
D = log2N (Length of a key tree)
Kij = Node key
i = Depth of a tree
j = Node key from the left
Kij = Key before rekeying
K'ij =Node key after rekeying

4 Suggested A-Key Management Methods

4.1 Modbus serial communications

This study reviews the Unicast mode and the Broadcast mode as the environment of RS-485.

Uncast mode	Boradcast Mode
Master: Request(Query)	Master: All Slaves Requests
Slave : Reply(Response)	
Address: 1 ~ 247	Address: 0

Table 4. Comparison of RS-485 Mode

In this study both individual rekeying and batch rekeying are examined, taking into account frequent changes in the client environment. That is, individual keys are distributed as long as the method causes no excessive load and, if not, batch rekeying is used to transmit/receive data through the broadcast communication. Initially, keys are distributed through 1:1 communication by Unicast and, if three or more of RTU request data, batch rekeying is used, converting to the broadcast mode so as to lower communication and master system loads.

4.2 Initial key distribution

① The initial A-key is an individual key and has a certified public key according to the following structure. When there is a request for communications, the certified public key is sent to an RTU. This is a fixed key value within a set according to the initial secret key set cycle for generation of a volatile secret key, which is later rekeyed. That is, it has the initial set key, which is used as a basic key for creating a secret key as shown in figure 7.

② A volatile secret key is created through a given secret key and a received RTU key as shown in figure 8.

③ Using the created volatile secret key, data are encoded and transmitted/received to make sure if the same key is shared as shown in figure 9.

This is how the initial key is created in the unicast environment. The key creation algorithm mentioned above is explained in detail in 4.3.

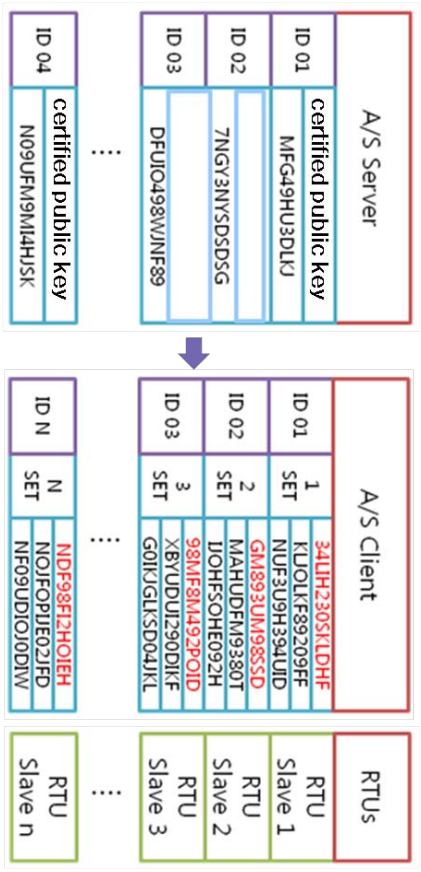

Fig.7 Initial Key Distribution Step 1

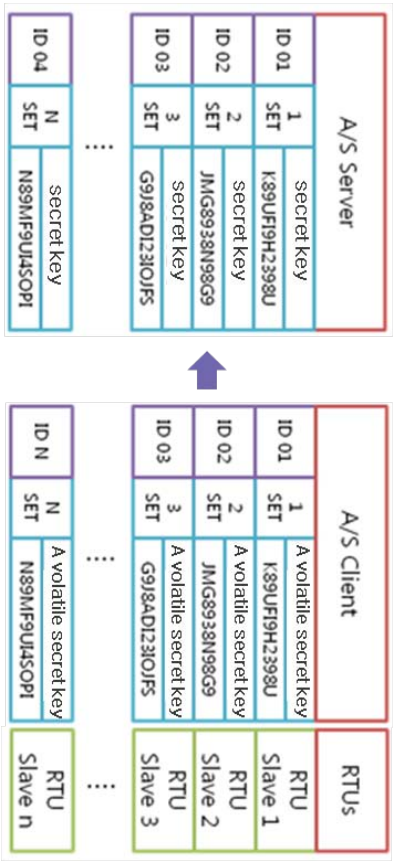

Fig.8 Initial Key Distribution Step 2

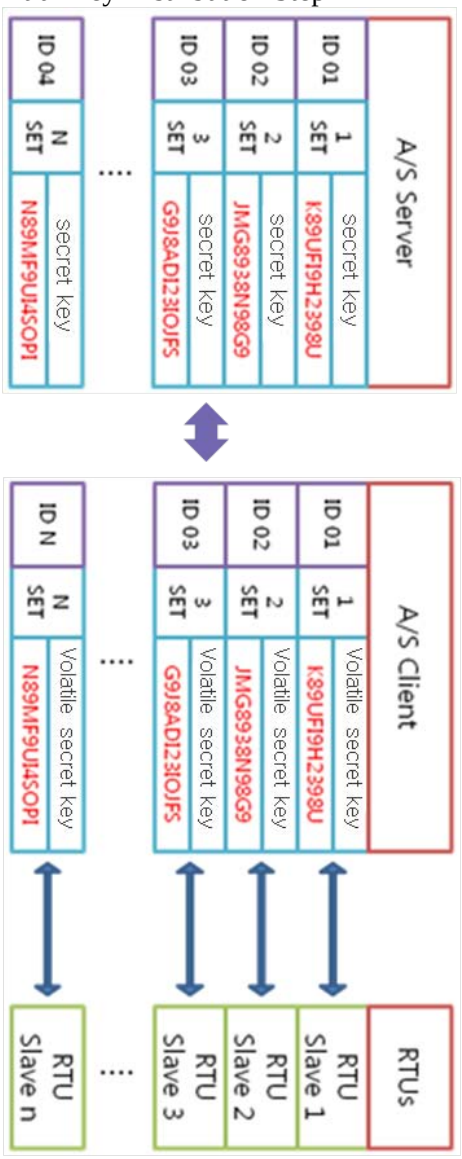

Fig.9 Initial Key Distribution Step 3

4.3 Secret key creation algorithm

In this study, a secret key is generated along with Master ID and RTU Client ID/Key in a reliable and safe way.

The major key authentication method suggested in this study is as follows.

EKd = EKd(RtuM_id || RtuS_id || Kr) - 256bit EKd1 = H(EKd ||01128)–128bit EKd2 = H(EKd ||02128)–128bit

The method described above uses EKd1 to confirm if the same key has been generated through EKd1/EKd2, generated in a way similar to SKKE, and uses EKd2 as a secret key.

The steps depicted above use a reliable secret key such as SKKE protocols in order to create a new trustworthy relationship[23].

The unique conditions of the industrial infrastructure require a stable and effective key distribution system to guarantee adequate transmission speed. To that end, the SEED algorithm, approved by the international standards, is employed.

The following figure 10 briefly describes the key distribution process.

▪Confirms authentication/Interrupts communications in case of failure of authentication
▪Generates and transmits a broadcast key
▪Starts encryption communications

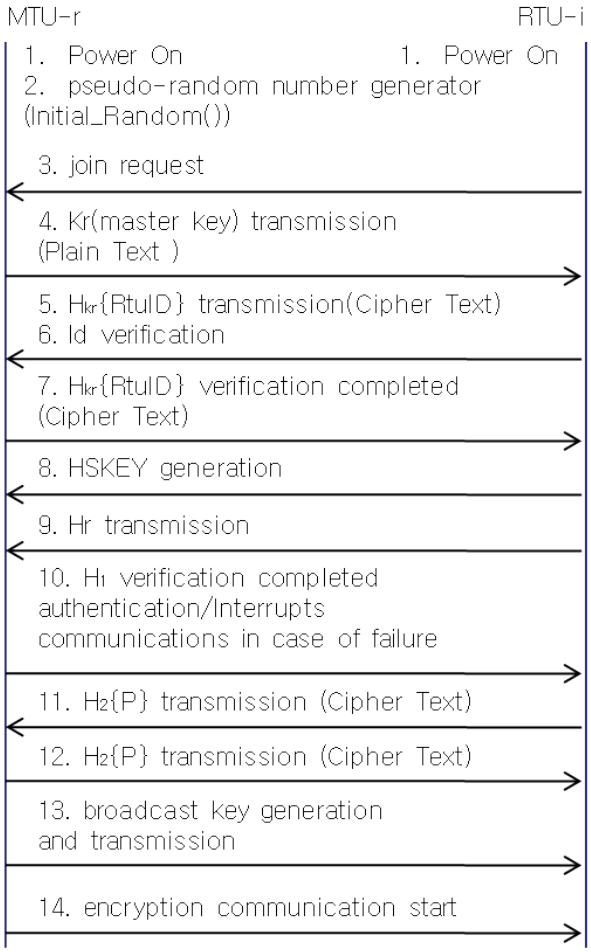

Fig.10 Flow of Key Distribution

① Regarding MTU and RTU, key distribution is carried out, supposing that the power is on.

② MTU renews a random number every 10 minutes using the function of Initial_Random().

③ RTU sends a request for joining to MTU and a communication starts.

④ MTU, using the created random number, generates a master key and sends it back to RTU.

⑤ RTU encodes its id based on the master key and sends it to MTU.

⑥ MTU decodes it and compares it with the RTU id saved in its list to verify the access.

⑦ MTU encodes its id(eigenvalue) to send it back to the authenticated RTU.

⑧ MTU and RTU create A-key H1 using their own id (MUT_id, RTU_id).

⑨ RTU sends H1 to MTU.

⑩ MTU verifies H1 received from RTU. If the value is correct, transmission/receiving starts through 1:1 by the unicast mode. If not, the communication stops and Step 4 is repeated.

⑪ Data are encoded using H1 and sent back to MTU.

⑫ MTU also encodes data using H1 key and transmit them to RTU.

⑬ If more than one RTU request authentication and, no problem is found up to 10 times, MTU creates a broadcast key using Initial_Random() and encodes it using H1, H2, H…in order to send it to a number of RTU.

⑭ A number of RTU receiving the broadcast key from MTU start a broadcast communication with MTU.

4.4 Block encryption algorithm

Based on the key distribution method designed through A-key management method, the SEED algorithm is applied in order to encode data. The SEED algorithm has the following features.

	SEED	ARIA
Block size	128 bit	128 bit
Length	128bit (Fixed key)	128,192,256bit (variable key)
Structure	Feistel	Involutional SPN
Round count	16	12/14/16 (variable size)

Table 5. Variable Characteristics of SEED and ARIA

5 System Design

5.1 Hardware design

The hardware of the system is composed of MTU, RTU, and KSC as shown in figure 11. In this study, it is supposed that the function of MTU is not affected even if it includes KSC (Key Set Center) and is physically safe also. The key distribution algorithm suggested in this study is included in KSC.

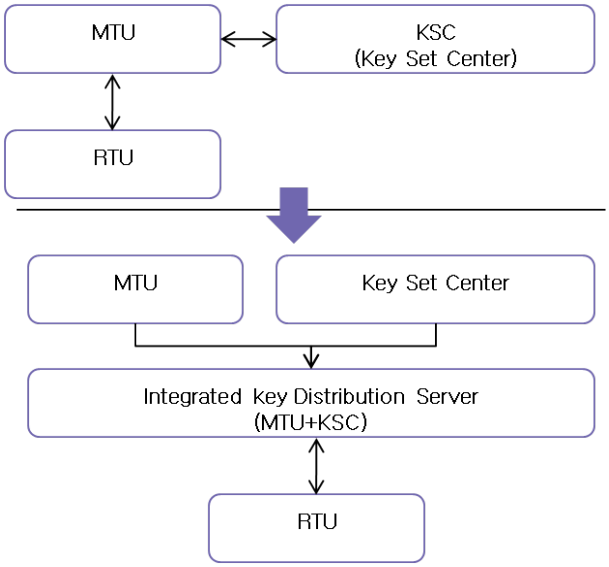

Fig.11 Hardware design

5.2 Software design

The key distribution system developed in this study is composed of S/W, controlling key distribution processes though physical connection between RTU Master and Client. The following figure 12 shows the structure of key distribution system connection.

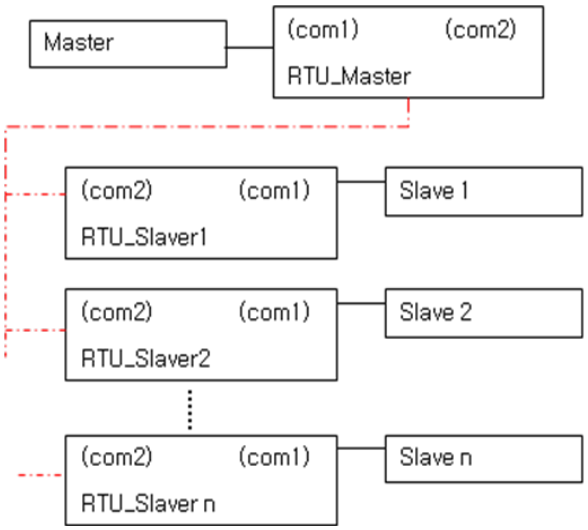

Fig.12 Key Distribution System Connection

In order to apply this system to actual H/W, the dotted lines should be at least 4 times greater than the undotted lines for stable key distribution. Message formats include STX, T_Length, Destination, Source, Command, D_Length, Data, ChecKSum, and ETX as follows.

① STX : 0x02

② From a destination to CheckSum

③ Address of a destination

④ Address of a starting place

⑤ Sequence no. of a transmission device(number in the box of the following page, ex: 3-SID transmission)

⑥ Length of the original data (2byte)

⑦ Transmission of data (ASCII code) 0xAA based on nibble-unit conversion, transmission of 0x41 0x41

⑧ Xor value(2 byte) between Data 1 ~ Data N

This system has Modbus communication structure of Unicast(1:1) and Broadcast(1:n) and the key distribution mechanism has the following purpose.

① To be effective in terms of storage space for encryption key management, arithmetic overhead, communications overhead, and so on (considering time between key generation and update)

② To make possible secure network through key update

This system generates pseudo-random numbers through SHA256, optimized in 32bit CPU. In case the first RTU Client attempts to log on, a connection is made through Unicast and when Client attempts to log on later, a key is generated through Broadcast. Figure 13 describes data encrypted through key distribution.

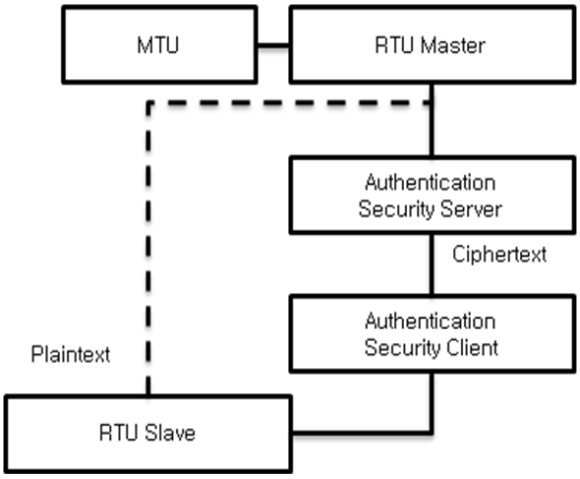

Fig.13 Data Flow

In order to prevent random access of unknown users who attempt to find out an encryption key, if a communication between Master and Client fails 5 times or more, the operation goes back to generation of pseudo-random numbers to restart distribution. If this process repeats 5 times or more, a communication failure message appears, interrupting key distribution and authentication process. In addition, an idle frame is transmitted per second after the connection between Master and Client has been completed and, if a response is not confirmed 5 times or more, reading and writing is interrupted.

The average time needed for encryption/description in this system, measured by GetTickCount system timer as shown in Table 6.

Plaintext	Ciphertext	Decryption
0.21ms	0.19ms	0.20ms

Table 6. Average time of plain text encryption/description

The average time needed for a plain text to be sent from RTU Master to Client is 0.21 ms and the total time including encryption/description is as follows.

0.18 + 0.20 + 0.21 = 0.56ms

Although there is about 37.5% overhead compared to 1:1 and 1:N communications, system capacities may cause variations. And even if each key is updated through Broadcast with fully connected RTU Client, about 150ms speed is necessary for one-time transmission, indicating that overhead of RTU Master is negligible.

6 Conclusion

This research analyzes characteristics of RS-485 protocols and suggests a key distribution method to protect data during data transmission of the industrial infrastructure. With this method, data cannot be decrypted even if exposed, for the data are certified through a secret key, different from previous transmission of a plain text. Furthermore, although the speed is slightly slower than previous plain text transmission, it will not affect the entire system and the increase in transmission time is negligible. This will cause no significant problem when the method is applied to an actual system.
Also, to solve the problem of A-key update when there is a great number of Clients, Broadcast key distribution method is applied, maintaining competitive performance.

Follow-up research shall be done in order to apply this system not only to Modbus but also to a DNP3 environment, as well as developing various security measures to deal with vulnerabilities arising from connection with TCP/IP protocols.

Corresponding author
Seoksoo Kim(sskim0123@naver.com)

References:

[1] The Office of Energy Assurance for the U.S. DoE, *21 Steps to improve Cyber Security of SCADA Networks*, 19, Sep. 2002.
[2] Jonathan Pollet, Safety Considerations for SCADA/DCS Cyber Attacks, *ISA-The Instrumentation Systems, and automation Society*, 1. Nov. 2003.
[3] White House, *The National Strategy to Secure Cyberspace*, Feb. 2003.
[4] Cepisca, Costin, Andrei, Horia, Petrescu, Emil, Privu, Cristian, Petrescu, Camelia, *Remote Data Acquisition System for Hydro Power Plants, 6th World Scientific and Engineering Academy and Society International Conference on Power Systems*, 2006.
[5] V Gaitan, V Popa, I Ungurean, NC Gaitan, The integration of real device capabilities in distributed applications based on OPC tehnology, *12th World Scientific and Engineering Academy and Society International Conference on COMPUTERS*, 2008.
[6] HEUNG S, DUTERTRE B, FONG M, LINDQVIST U, SKINNER K, AND VALDES A,Using model-based intrusion detection for SCADA networks, *In Proceedings of the SCADA Security Scientific Symposium*, 2007.
[7] *Modbus Protocol Reference Guide*, www.essproducts.com/serial_communications.
[8] J. Stanek, *Introduction to RS 422 & RS 485*, http://www.hw.cz.
[9] Trexon Inc., *Modbus Protocol*, Jan. 2000.
[10] MODICON, Inc., *Modicon Modbus Protocol Reference Guide*, June. 1996.
[11] National Institute of Standards and Technology., *Information Security*, May, 2008.
[12] Cao Chunjie, Ma Jianfeng and Moon Sangjae., Provable efficient certificateless group key exchange protocol, *Wuhan University Journal of Natural Sciences*, 2007, pp 41-45.
[13] Rakesh Bobba and Himanshu Khurana, DLPKH, Distributed Logical Public-Key Hierarchy, *Information systems security: third international conference* , 2007, pp 110-127.
[14] J. Castillo,P. Huerta,J. I. Martinez,Juan Carlos, SystemC design flow for a DES/AES CryptoProcessor, *World Scientific and Engineering Academy and Society (WSEAS)*, 2004.
[15] Alex Biryukov, Christophe De Canni`ere, Joseph Lano, Siddika Berna Ors, Bart Preneel, *Security and Performance Analysis of Aria, Version 1.2. 7*, Dept. Electrical Engineering-ESAT/SCD-COSIC Katholieke Universiteit Leuven Kasteelpark Arenberg 10, B–3001 Heverlee,2004.
[16] Nor Badrul Anuar, Lai Ngan Kuen, Omar Zakaria, Abdullah Gani, Ainuddin Wahid Abdul Wahab, GSM mobile SMS/MMS using public key infrastructure: m-PKI, *World Scientific and Engineering Academy and Society (WSEAS)*,Volume 7 , Volume 8, 2008, pp. 1219-1229.
[17] R. Alvarez, F. M. Martinez, J. F. Vicent, and A. Zamora, A Matricial Public Key Cryptosystem with Digital Signature, *World Scientific and Engineering Academy and Society*

Transactions on Mathematics, Issue 4, Volume 7, 2008, pp.195-204.

[18] Cao, C. J., Ma, J.F, Identity-based Constant Round Group Key Exchange Protocol via Secret-Share. *Proceedings of World Scientific and Engineering Academy and Society TRANSACTIONS on SYSTEMS*. January 2008.

[19] S. Banerjee and B. Bhattacharjee, Scalable secure group communication over IP multicast, *IEEE J-SAC*, Vol.20, No.8,2002, pp.1511-1527.

[20] Wan Fang, Wang Dazhen, An group key distribution protocol for secure group communications, *6th World Scientific and Engineering Academy and Society International Conference on Applied Computer Science*, 2007.

[21] H. Harney and C. Muckenhirn, Group Key Management Protocol (GKMP) Architecture, *Internet Engineering Task Force*, 1997.

[22] Waldemar Grabski, Micha l Nowacki, Code Generation for CSM/ECSM Models in COSMA Environment, *International Multiconference on Computer Science and Information Technology*, 2006, pp. 393-400.

[23] Moazzam Khan, Fereshteh Amini and Jelena Mišić, Key Exchange in 802.15.4 Networks and Its Performance Implications, *Springer Berlin*, 2007, pp 497-508.