International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:9, 2008

An Efficient Key Management Scheme for Secure SCADA Communication

Sungjin Lee, Donghyun Choi, Choonsik Park, and Seungjoo Kim

***Abstract*—**A SCADA (Supervisory Control And Data Acquisition) system is an industrial control and monitoring system for national infrastructures. The SCADA systems were used in a closed environment without considering about security functionality in the past. As communication technology develops, they try to connect the SCADA systems to an open network. Therefore, the security of the SCADA systems has been an issue. The study of key management for SCADA system also has been performed. However, existing key management schemes for SCADA system such as SKE(Key establishment for SCADA systems) and SKMA(Key management scheme for SCADA systems) cannot support broadcasting communication. To solve this problem, an Advanced Key Management Architecture for Secure SCADA Communication has been proposed by Choi et al.. Choi et al.'s scheme also has a problem that it requires lots of computational cost for multicasting communication. In this paper, we propose an enhanced scheme which improving computational cost for multicasting communication with considering the number of keys to be stored in a low power communication device (RTU).

***Keywords*—**SCADA system, SCADA communication, Key management, Distributed networks.

I. Introduction

A SCADA (Supervisory Control And Data Acquisition) system is an industrial control and monitoring system for various national infrastructures. In the past, the SCADA systems operated in a closed environment and the security of the systems was not considered. Therefore, the SCADA systems become vulnerable if they are connected to an open network. Nowadays, as a demand for connecting the SCADA system to the open network increases, the study of security for the SCADA systems has been an issue and encouraged. Several key management schemes for the SCADA systems also have been proposed already.

"This research was supported by the Ministry of Knowledge Economy, Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)" (IITA-2008-C1090-0801-0016)

"This research was supported by the MKE(Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)" (IITA-2008-C1090-0801-0028)

S. Lee, D. Choi, and S. Kim (Corresponding author) are with the Information Security Group,Sungkyunkwan University, 300 Cheoncheon-dong, Jangan-gu, Suwon-si, Gyeonggi-do, 440-746, Korea (phone: 82-31-290-5695; fax: 82-31-290-5696; e-mail: {sjlee, dhchoi, skim}@security.re.kr).

C. Park is with the Attached Institute of ETRI, Yuseong P.O.Box #1, Yuseong-gu, Daejeon, 305-600, Korea (phone: 82-42-870-2101; e-mail: csp@ensec.re.kr).

However, Key establishment for SCADA systems (SKE) [2] and Key management scheme for SCADA systems (SKMA) [3] have a disadvantage that they cannot support broadcasting communication. An Advanced Key Management Architecture for Secure SCADA Communication [1] has been proposed by Choi et al. to solve this problem through logical key hierarchy. However, Choi et al.'s scheme still has a problem that it requires lots of computational cost. This problem disturbs the real-time processing of the SCADA systems.

In this paper, we divide the key structure into two classes by applying Iolus framework and construct each class as a logical key hierarchy structure. Through this key structure, we propose an enhanced key management scheme supporting efficient multicasting. Moreover, our scheme is proposed with considering the number of keys to be stored in a low power communication device having limited resources.

In section II, we describe the structure of the SCADA system and requirements for key management scheme. Section III summarizes of related key management schemes. Then, we propose a key management protocol for the SCADA systems in section IV. In Section V, we compare our scheme with the existing key management schemes. Finally, Section VI concludes this work.

II. SCADA System

A. Structure of the SCADA Systems

A SCADA system monitors and controls remote facilities by collecting data from various sensors over the SCADA networks. SCADA systems have a hierarchical structure. The communication type of SCADA systems is also master-slave structure. Fig. 1 shows the simplified architecture of a SCADA system.

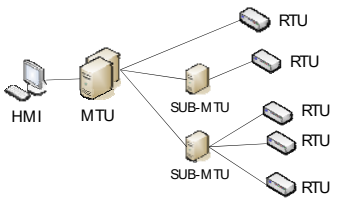


Fig. 1 Structure of the SCADA system

A SCADA system consists of three types of communication equipments which are HMI (Human Machine Interface), MTU (Master Terminal Unit) and RTU (Remote Terminal Unit) in a

simple expression [1].

The network topology of the SCADA systems is static. The communication paths between nodes are known in advance because there are only few changes in the network. The communication is occurred between HMI and MTU, MTU and SUB-MTU, two SUB-MTUs, MTU and RTU, SUB-MTU and RTU, and two RTUs. The HMI-MTU communication can be provided by the web service easily because of using TCP/IP based protocols. Moreover, the HMI–MTU communication has less resource limitation than the other communication.

B. System Constraints and Security Requirements

A SCADA system network is different from general network environment due to its operational environment in national infrastructure. Therefore, the system network has following constraints.

1) **Limited computational capacity**: The remote equipment such as RTUs is a embedded system having low computational and space capacity.
2) **Low rate data transmission**: Since the SCADA system has been used for a long time, the communication line of the SCADA network has low bandwidth.
3) **Real-time processing**: The SCADA system should behave accurately. Delay of data processing could cause serious problem.

Above constraints of SCADA systems make it difficult to apply the security technology to the system so that the constraints should be a basic consideration for applying security mechanism.

Key management schemes are used for confidentiality. The SCADA system need broadcast capability to send important messages such as “emergency shutdown” [1]. The target receiver of these messages can be all the communication equipments or specific multiple communication equipments in the SCADA systems. Therefore, multicasting should be required for secure SCADA communications. Key management schemes also have to support the multicasting to protect the multicast messages.

Furthermore, the number of keys to be stored in a RTU or a MTU should be minimized. Since the RTU has limited memory, the number of keys to be managed in the RTU should be considered. The efficiency of memory space in the MTU also ought to be considered because the MTU manages and maintains the data and keys from the thousands of RTUs in a limited memory space. Hence the number of keys to be stored in a RTU or a MTU has to be significantly reflected on the key management scheme for SCADA systems.

The computational cost for member join or member leave protocol of key management scheme is not important and ignorable since addition or deletion of member is rarely happened in the SCADA systems.

Consequently, the following requirements are essential for efficient key management scheme for secure SCADA communication.

- Supporting broadcasting and multicasting
- The number of keys to be stored in a RTU

III. Related Works

A. Existing Key Management Method for SCADA

Key establishment for SCADA systems (SKE) [2] was proposed by SANDIA. This mechanism does not support broadcasting communication. In other words, the MTU has to encrypt data with each key of RTUs individually to broadcast a message. Hence, it needs $O(n)$ computational cost when n is the number of RTUs. Moreover, SKE require two types of key to maintain which are LTK (Long Term Key), GSK(General Seed Key) and GK(General Key). A MTU has to maintain one LTK and one GK for one RTU and one GSK for itself. A RTU should store one LTK and one GK shared with the MTU. Consequently, a MTU stores $2n+1$ keys and a RTU stores two keys for communication.

Key management scheme for SCADA systems (SKMA) [3] also does not support broadcasting communication as SKE. It means it needs $O(n)$ computational cost for broadcasting a message. In SKMA, there are two types of keys to be managed by a MTU or RTUs. Long term node-KDC(Key Distribution Center) key is shared by between KDC and a node. Another key is long term node-node key which is shared between two nodes. The long term node-KDC key is distributed manually, while the long term node-node key is distributed by KDC. Hence, a RTU needs to store n keys as the same amount of keys stored in a MTU.

Choi et al. [1] proposed a key management scheme for SCADA systems using logical key hierarchy to support broadcasting communication. This scheme also provides multicasting communication between the MTU and multiple SUB-MTUs. However, this scheme cannot support multicasting communication between a MTU and multiple RTUs as well as between a SUB-MTU and multiple RTUs. The reason is that all the RTUs are attached to the one SUB-MTU as a star graph. Another drawback of this scheme is that the logical key hierarchy scheme requires the MTU and the RTUs to store more keys than the above schemes. Let m be the number of the SUB-MTUs and n be the number of the RTUs. A MTU has to store $2m-1+n$ keys which are all the keys of all nodes of binary tree having m leaf nodes and each key of n RTUs. A RTU has to store $2+\log_2 m$ keys which are one nonce value to generate a session key, its own secret key and all the keys of all the nodes in the path from the SUB-MTU to the MTU in the logical key hierarchy structure.

B. Centralized Group Key Agreement Protocol

Centralized group key agreement protocol is a group key agreement protocol between nodes in a group by using a centralized server managing the group. The basic centralized group key agreement protocol is Single Key Distribution Center (SKDC) [4]. In SKDC, the server transmits an encrypted group key with shared key between server and members to the members. SKDC is simple but has problem to apply to large group communication since the cost for managing the keys increases proportionally to the number of keys. In other words, total number of keys to be stored is n for

MTU and one for RTU.

Logical Key Hierarchy (LKH) [5] based on constructing a logical tree of keys has been proposed to solve the scalability problem. Disadvantage of this scheme is that the number of keys to be stored is increased. Total amount of the number of keys is $2m-1$ for the root node and $\log_2 m$ for a leaf node.

As a more enhanced method, One-way Function Tree (OFT) [6] has been proposed to reduce the cost for managing the keys in an half. In OFT, the parent node is computed by hashing the hashed values of its two child nodes. Therefore, the root node can generate a key for broadcasting and multicasting with storing only n keys. However, a leaf node should store $\log_2 m$ keys.

C. Iolus framework

Iolus framework [7] is a method to manage the secure distribution tree which of each node is a subgroup managed by a subgroup manager. There are two types of managers for managing group. One is GSC(Group Security Controller) which manages the top-level subgroup. Another one is GSI(Group Security Intermediary) which exists in a subgroup and manages the other subgroups except the top-level subgroup. The GSIs share group key with the GSC. Each node shares subgroup key with the GSI where the node belongs.

If a node wants to transmit a message to the other nodes, the node encrypts the message with a random number key K_r and encrypts the K_r with the subgroup key G_i and sends the encrypted message and key to the GSI. The GSI decrypts the key with G_i and re-encrypts with the group key G_j of another group and multicasts the message to other GSIs. The other GSIs decrypt the key with their group key G_j and re-encrypt with their own subgroup key G_k then multicast. Afterward, the only subgroup member knowing G_k can decrypt the key K_r and decrypt the message with the K_r.

IV. The Proposed Scheme

In this paper, we propose an efficient key management protocol for secure SCADA communications. As we mentioned above, the key management for SCADA system should consider the number of keys and provide broadcasting and multicasting communication. Our proposal satisfies these considerations by using logical key hierarchy and Iolus framework scheme. The logical key hierarchy can provide multicasting and broadcasting communication between one MTU and multiple SUB-MTUs, and one SUB-MTU and multiple RTUs. Furthermore, our scheme applies the Iolus framework to reduce the number of keys to be stored. The Iolus framework is appropriate for the SCADA systems because the SUB-MTU relays messages between a MTU and a RTU in the SCADA communications. Therefore, in our scheme, the SUB-MTU takes a role of a GSI and the role of MTU is a GSC.

A. Notation and Definitions

The following notations are used throughout this paper.

B. Initialization

TABLE I
Notations

Symbol	Quantity
h	height of a tree
m	the number of SUB-MTUs
n	the maximum number of RTUs per SUB-MTU
N_{MT_i}	number of the RTUs attached to i_{th} SUB-MTU node MT_i
KDC	key distribution center in a MTU
MT_i	i_{th} SUB-MTU node of current SUB-MTU group. $1 \le i \le m$
MT_0	the MTU node
MT	set of current SUB-MTU members, $MT = \{MT_0, MT_1, \ldots, MT_m\}$
RT_i	i_{th} RTU node of current RTU group. $1 \le i \le n$
RT	set of current RTU members, $RT = \{RT_1, RT_2, \ldots, RT_{mn}\}$
$K_{i,j}$	j_{th} auxiliary key at level i in a binary tree for MT set, $K_{0,1}$ is a root key of the binary tree. $0 \le i \le h$, $1 \le j \le m$
$K^s_{i,j}$	j_{th} auxiliary key at level i in a binary tree for RT subset of MT_s. $1 \le s \le m$, $1 \le i \le \log_2 n$ and $1 \le j \le n$
K_i	secret key of i node
TVP	combination of timestamp and sequence number
$E_K(D)$	encryption function of a symmetric cipher using a secret key K
$H(D)$	hash function
$SK_{i,j}$	a session key between i node and j node
A→ B:{Msg}	A sends a message Msg to B. A and B can be a set of entities. The set of entities is expressed to { entities }
$\min(i, j)$	the minimum number between i and j

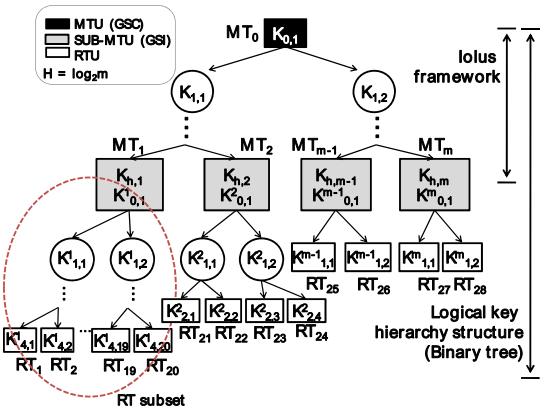


Fig. 2 Key structure of our scheme

The KDC constructs logical key structure and uses Iolus framework as shown in Fig. 2. The structure has two sets, MT and RT. The key structure for each set is constructed as a logical key hierarchy structure. The key structure of MT set and RT set is connected through Iolus framework. The MTU plays a role of GSC and manages the entire group and group key between the MTU and SUB-MTUs. The SUB-MTU plays a role of GSI. It manages the subgroup key of its subgroup consisting of n RTUs and relays messages between RTUs and MTU or RTUs and other SUB-MTUs.

Key structure of MT set. The keys for MT set are organized as a binary tree with total $2m-1$ nodes so that the height

$h=\log_2 m$ of the tree is determined by m which is the number of SUB-MTUs.

In the key structure, the keys of leaf nodes MT_i $(1\le i\le m)$ are $K_{h,j}$ $(1\le j\le m)$ and assigned to all SUB-MTUs. The other keys $K_{i,j}$ $(1\le i\le h-1, 1\le j\le m)$ are generated by following equation. In other words, the key is generated by hashing the hashed values of its child nodes.

$$K_{i+1,\lceil j/2\rceil}=H(H(K_{i,j}),H(K_{i,j+1}))$$
$$(1\le i\le h-1, 1\le j\le m)$$

The key of the root node MT_0 is $K_{0,1}$ and generated from above equation. As a result, the MTU can know the root key by containing only $K_{h,j}$ $(1\le j\le m)$ of m SUB-MTUs.

Key structure of RT subset. Each RT set is organized as a binary tree with $2(N_{MT_i})-1$ nodes so that each RT set has a different height $h=\log_2(N_{MT_i})$.

In the key structure of RT subset managed by MT_i, the keys of leaf nodes RT_j $(1\le j\le n)$ are $K^{i}_{\log_2 N_{MT_i},j}$ and assigned to all the RTUs. The other keys $K^{s}_{i,j}$ $(1\le s\le m, 1\le j\le n, 1\le i\le \log_2 n)$ are generated by hashing the hashed values of its child nodes like following equation.

$$K^{s}_{i+1,\lceil j/2\rceil j}=H(H(K_{i,j}),H(K_{i,j+1}))$$
$$(1\le i\le \log_2 N_{MT_s}-1,\quad 1\le j\le n)$$

A root key $K^{i}_{0,1}$ of RT subset managed by MT_i and generated from the above equation. As a result, the SUB-MTU with node MT_i can know the root key by containing only $K^{i}_{\log_2 N_{MT_i},j}$ $(1\le j\le n)$ of n RTUs.

C. Join Protocol

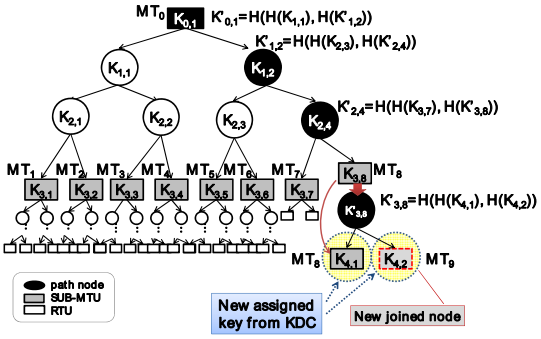


Fig. 3 Member join protocol in our scheme

To preserve backward secrecy in join protocol, all keys that a new RTU or a new SUB-MTU receives need to be independent from any previous keys. Hence, in the case of SUB-MTU join protocol, the KDC replaces all the keys on the new SUB-MTU's key path. In the case of RTU join protocol, the SUB-MTU replaces all the keys on the new RTU's key path of the RT subset. Therefore, each join protocol in a group does not influence on the other group set. The join protocol uses a hash function to update keys.

SUB-MTU join protocol. When a new SUB-MTU is added to the SCADA system, the key structure for MT set is changed and new group key for SUB-MTUs is distributed to the SUB-MTUs according to the following procedure.

1) A new SUB-MTU (MT_{m+1} node) is assigned a new key $K_{MT_{m+1}}$ from the KDC.
2) The new node MT_{m+1} is inserted to the key structure of MT set as a sibling node of where it is inserted to.
3) The KDC updates the old key K_{MTm} of the sibling node (MT_m) of the new node to a new key K'_{MT_m} by sending encrypted the new key K'_{MT_m} to the sibling node.
$$KDC\rightarrow MT_m:\{E_{K_{MT_m}}(K'_{MT_m})\}$$
4) The KDC updates all the keys of the path nodes from the new node to the root node. The updated key of a path node is generated by hashing the hashed values of the keys from its child nodes. All the key of all the nodes of the path from the parent node of the new node to the root node in Fig. 3 is updated as following computation.
$$K'_{3,8}=H(H(K_{4,1}),H(K_{4,2}))$$
$$K'_{2,4}=H(H(K_{3,7}),H(K'_{3,8}))$$
$$K'_{1,2}=H(H(K_{2,3}),H(K'_{2,4}))$$
$$K'_{0,1}=H(H(K_{1,1}),H(K'_{1,2}))$$
5) The KDC multicasts the updated keys by encrypting the key with its old key and unicasts them to the new node.
$$KDC\rightarrow\{MT\}:\{E_{K_{i,j}}(K'_{i,j}),...\}$$
$$KDC\rightarrow MT_{m+1}:\{E_{K_{MT_{m+1}}}(K'_{i,j},...)\}$$
6) The KDC unicasts the key of the parent node of the new node to the sibling node of the new node. In Fig. 3, the unicasting message is generated and transmitted like following equation.
$$KDC\rightarrow MT_8:\{E_{K_{MT_8}}(K'_{0,1},K'_{1,2},K'_{2,4},K'_{3,8})\}$$

RTU join protocol. RTU join protocol performs the same procedure as SUB-MTU join protocol replacing the KDC to the SUB-MTU and the new SUB-MTU to the new RTU.

D. Leave Protocol

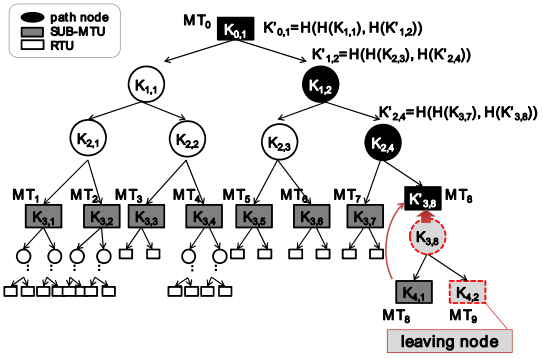


Fig. 4 Member leave protocol in our scheme

The goal of this leave protocol is that leaving RTU or SUB-MTU must not be able to compute new keys. Therefore, all the keys known by the leaving RTU or SUB-MTU have to be updated.

SUB-MTU leave protocol. When a SUB-MTU is evicted from the SCADA system, the key structure for MT set is changed and new group key for remained SUB-MTUs is distributed to the SUB-MTUs according to the following procedure.

1) The leaving SUB-MTU(MT_m node) is deleted from the MT set key structure. Then the sibling of the leaving node (MT_{m-1} node) is reassigned to the parent of the node.
2) MT_{m-1} is given a new key $K'_{MT_{m-1}}$.
3) The KDC updates all the keys of the path nodes from MT_{m-1} to the root node. The updated key of a path node is generated by hashing the hashed values of the keys from its child nodes.
4) The updated key of each path node and its high level path nodes' updated keys are encrypted with the key of its child node not in the path then multicast. The key of the path node in Fig. 4 is updated as following computation.

$$K'_{2,4} = H(H(K_{3,7}), H(K'_{3,8}))$$
$$K'_{1,2} = H(H(K_{2,3}), H(K'_{2,4}))$$
$$K'_{0,1} = H(H(K_{1,1}), H(K'_{1,2}))$$

5) The KDC encrypts all the key of the path nodes with $K'_{MT_{m-1}}$ and unicasts the message to the MT_{m-1}.

RTU leave protocol. RTU leave protocol performs the same procedure as SUB-MTU leave protocol replacing the KDC to the SUB-MTU and the leaving SUB-MTU to the leaving RTU.

E. Data Transmission

In our scheme, not only node to node communication but also broadcasting or multicasting communication can be provided. The hierarchical structure of logical key can organize the multiple nodes in a group dynamically and make several subgroups in a group. Consequently, multicasting and broadcasting communication between one MTU and multiple SUB-MTUs, one SUB-MTU and multiple RTUs, and one MTU and multiple RTUs is achieved in our scheme.

In the SCADA systems, the encryption key should be refreshed regularly, because there are no known ways of automatically discovering a key compromise. Besides, the key freshness prevents the replay attack. To guarantee key freshness, all the encryption keys are replaced with new key for each session. Therefore, our mechanism uses a TVP which is a combination of timestamp and sequence number.

The basic data flow is as the following explanation. Since our scheme follows Iolus framework, any messages sent by a MTU should pass the SUB-MTU. First, the MTU generates a random key and encrypts the messages with the random key. Then, the MTU encrypts the random key with the session key shared between the MTU and the one or multiple SUB-MTUs. When the SUB-MTUs receive the encrypted messages and the encrypted random key, the SUB-MTUs decrypt the random key and re-encrypt with another session key which is shared with the one or multiple RTUs and multicast the re-encrypted random key to the RTUs. Therefore, only the selected RTUs are able to decrypt the messages with the random key. In the case of that the RTU sends messages, the messages should pass the SUB-MTU which manages the RTU. The messages flow to the MTU according to the above procedure.

Specifically, the communication type of the SCADA system is divided into two which are Node-to-Node communication and Node-to-Group communication. The Node-to-Node communication can be occurred between a MTU and a SUB-MTU, a SUB-MTU and a SUB-MTU, a SUB-MTU and a RTU, and a RTU and a RTU. The Node-to-Group communication can be occurred between a MTU and multiple SUB-MTUs, a SUB-MTU and multiple RTUs and a MTU and multiple RTUs.

The Node-to-Node communication and the Node-to-Group communication have a different method for generating a session key. Basically, the session key consists of the shared key between the communication entities, ID of the two communication entities and TVP. The shared key is for generating the common key for both nodes. The ID of the node is used to prevent from generating the similar session key because the multiple pair of nodes could have the same shared key. Lastly the TVP protects the session key against the replay attack.

Node-to-Group communication. The session key for multicasting communication should be shared by all the members in *j* group as well as *i* node. Any pair of two nodes in the key structure share one or more keys. Therefore, *i* node and the members of *j* group are able to know the shared key between them in common. The session key generation method is like the following equation.

$$SK_{i,j} = H(K_{u,v}, TVP)$$

The key $K_{u,v}$ is the shared key between all the members of *j* group and *i* node. Through the logical key hierarchy structure, the multiple nodes can share a same key.

Node-to-Node communication. In the Node-to-Node communication, the session key generation method between *i* node and *j* node is explained as the following equation and description.

$$SK_{i,j} = H(K_{u,v}, ID_i, ID_j, TVP)$$

The key $K_{u,v}$ is the shared key between *i* node and *j* node through the logical key hierarchy structure. The session key generation method for node to node communication is different from the session key generation method of the Node-to-Group communication. The shared key is known by the other nodes in the group which is the two nodes belonging in common. Therefore, the session key should include the unique identities of the two nodes in order not to be known by the others.

After sharing the above session key between i node and j node, they communicate with each other directly.

V. Comparison

In the SCADA system, the multicasting communication from a MTU to multiple RTUs is needed for efficient communication. Existing key management schemes do not supports the multicasting. In our scheme, the multicasting

communication is achieved by using the logical key hierarchy structure in RT subset. Table II presents the supporting communication of each key management scheme.

TABLE II
THE SUPPORTING COMMUNICATION

	SKE or SKMA	Choi et al.'s scheme	Our scheme
Broad-casting	Not supporting	Supporting	Supporting
Multi-casting	Not supporting	Supporting	Supporting

The advantage of our scheme is to support more efficient multicasting communication than the other scheme. In our scheme, MTU or SUB-MTU is able to multicast messages efficiently through the key structure between a SUB-MTU and RTUs organized as a logical key hierarchy structure. It can improve the computational cost for data transmission during operation time. It is a critical advantage in real-time system like the SCADA system. Table III shows the computational cost for multicasting a message from a MTU to p SUB-MTUs in the MTU and from a SUB-MTU to q RTUs in the SUB-MTU. The SUB-MTU computes q times of message encryption in SKE or SKMA and Choi et al.. q can be a big number in a SCADA system managing thousands of RTUs. On the other hand, our scheme only needs one decryption of an encrypted key and Y times of re-encryption of the key in the SUB-MTU. X and Y are the number of keys for encrypting the multicasting message in the MTU or the SUB-MTU and determined by the location of the nodes of the selected SUB-MTUs or RTUs in the key structure. Therefore, X and Y are never over $n/2$, p or q through the form of the key structure as a binary tree.

TABLE III
COMPUTATIONAL COST FOR MULTICASTING

	SKE or SKMA	Choi et al.'s scheme	Our scheme
MTU	pC_E	XC_E	$C_E + XC_{E_K}$
SUB-MTU	qC_E	$(q \bmod (n-1))C_E$	$(Y+1)C_{E_K}$

C_E = computational cost for message encryption, C_{E_K} = computational cost for encryption of a key, p = the number of SUB-MTUs to receive the multicasting message from a MTU and $1 \le p \le m$, q = the number of RTUs to receive the multicasting message from a SUB-MTU and $1 \le q \le n$, X = the number of keys for encrypting the multicasting message in the MTU and $1 \le X \le \min(m/2, p)$, Y = the number of keys for encrypting the multicasting message in the SUB-MTU and $1 \le Y \le \min(n/2, q)$

Fig. 5 is the graph showing a computational cost for multicasting communication in a MTU according to the number of multicasting target SUB-MTUs with maximum 1000 SUB-MTUs. Fig. 6 is the graph showing a computational cost for multicasting communication in a SUB-MTU according to the number of multicasting target RTUs with maximum 1000 RTUs. In Fig. 5-6, C_E is the message encryption cost with a 1Mbit message and C_{E_K} is the key encryption cost with a 128 bits key.

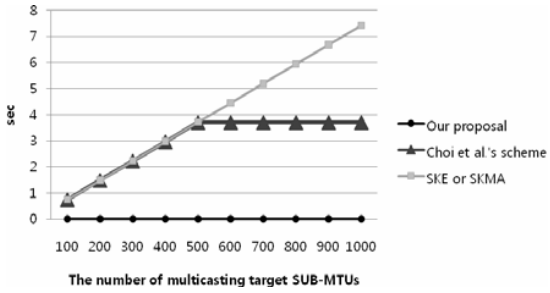


Fig. 5 Computational cost in a MTU according to the number of multicasting SUB-MTUs

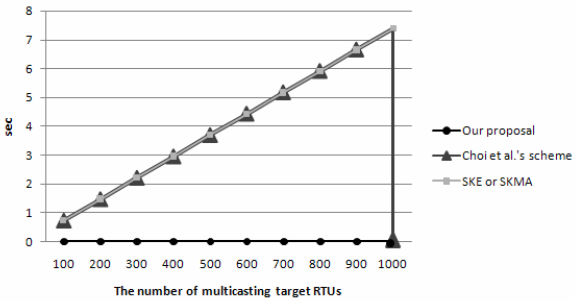


Fig. 6 Computational cost in a SUB-MTU according to the number of multicasting RTUs

Our scheme divides the key structure into two classes which are between MTU and SUB-MTUs, and between SUB-MTUs and RTUs through Iolus framework. Each class organizes the key structure as a logical key hierarchy structure. Thus, in our scheme, the key structure between SUB-MTUs and RTUs is a logical key hierarchy structure different from Choi et al.'s scheme. Even if the key structure between SUB-MTUs and RTUs is changed to the logical key hierarchy structure in Choi et al.'s scheme without Iolus framework, the number of keys to be stored in a RTU increases to $1+\log_2 mn$. Thus, we apply the Iolus framework to our scheme for preventing the number of keys to be stored in a RTU from increasing. The number of the keys is $1+\log_2 n$ in our scheme.

In our scheme, the number of keys to be stored in a SUB-MTU is more than in the other schemes. However, it can be ignorable because the SUB-MTU has similar computation power and resources with the MTU. Besides, the number of keys is more significantly considerable in a MTU and a RTU according to the key management requirements analyzed in the above. In our scheme, the number of keys to be stored in a MTU and a RTU is less than in the other schemes. Hence, our scheme is more efficient than the other schemes in the memory efficiency point of view

In Table IV, we compare the number of keys to be stored in a RTU for SKE or SKMA, Choi et al.'s scheme and our scheme.

TABLE IV
THE NUMBER OF KEYS TO BE STORED

	SKE or SKMA	Choi et al.'s scheme	Our scheme
MTU	mn	$2m-1+mn$	m
SUB-MTU	$1+n$	$1+\log_2 m$	$1+n+\log_2 m$
RTU	1	$2+\log_2 m$	$1+\log_2 n$

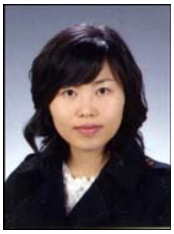
VI. CONCLUSION

A SCADA system is significantly important system used in national infrastructures such as electric grids, water supplies, and pipelines. However, the SCADA systems have lots of security vulnerabilities. Any faults or damages of the SCADA system can affect to the society severely. The study of the security for SCADA system is essential for that reason.

Specifically, our study focuses on the key management scheme for supporting data protection. We analyzed the requirements and constraints of SCADA system and its communication network at first. Afterwards, we proposed more efficient and secure key management scheme than existing schemes. Our scheme also solves the problem of the existing schemes. After all, our scheme the reduced the number of keys to be stored and provides multicasting and broadcasting communication for efficient and stable operation of SCADA systems.

REFERENCES

[1] Donghyun Choi, Hakman Kim, Dongho Won, and Seungjoo Kim, "Advanced Key Management Architecture for Secure SCADA Communication," To be published on IEEE Transactions on power delivery
[2] Beaver, C., Gallup, D., Neumann, W. & Torgerson, M. (2002), "Key management for SCADA," Technical report, Sandia. http://www.sandia.gov/scada/documents/013252.pdf
[3] Robert Dawson, Colin Boyd, Ed Dawson, Juan Manuel Gonzalez Nieto, "SKMA A Key Management Architecture for SCADA Systems," In Proc. Fourth Australasian Information Security Workshop, Vol. 54, pp. 138-192, 2006.
[4] Balenson, D., McGrew, D, and A. Sherman, "Key management for large dynamic groups: one-way function trees and amortized initialization," NAI Labs, Advanced Security Research Journal, pp 29-46. 1998
[5] Chung Kei Wong; Gouda, M.; Lam, S.S., "Secure group communications using key graphs," IEEE/ACM Transactions on Networking, vol.8, no.1, pp.16-30, Feb 2000
[6] McGrew, D.A.;Sherman, A.T., "Key Establishment in Large Dynamic Groups: Using One-Way Function Trees", Technical Report 0755
[7] S. Mittra, "Iolus: A Framework for Scalable Secure Multicasting," in Proc. ACM SIGCOMM'97, pp. 277-88, 1997



Sungjin Lee received her B.E. degree in Computer Engineering from Sungkyunkwan University, Korea, in 2007. She is currently a M.S. course of Mobile systems engineering in Sungkyunkwan University. Her current research interest is in the area of cryptography, SCADA and mobile security.

Donghyun Choi received his B.E. degree in Electrical and Computer Engineering from Sungkyunkwan University, Korea, in 2005 and M.S. degree in Computer Science from Sungkyunkwan University, Korea, in 2007. He is currently a Ph.D. course of Mobile systems engineering in Sungkyunkwan University. His current research interest is in the area of cryptography, SCADA, mobile security, and DRM.

Choonsik Park received his B.E. degree from Kwangwoon University, Korea in 1980, M.E. degree from Hanyang University, Korea in 1982, and Ph.D. degree from Tokyo Institute of Technology, Japan in 1995. He has been working at ETRI (Electronics & Telecommunications Research Institute) since 1982. His current research interest is in the area of cryptography and mobile security.

Seungjoo Kim received his B.S. (1994), M.S. (1996), and Ph.D. (1999) in information engineering from Sungkyunkwan University (SKKU) in Korea. Prior to joining the faculty at SKKU in 2004, he had an appointment as the Director of the Cryptographic Technology Team & the (CC-based) IT Security Evaluation Team of the Korea Information Security Agency (KISA) for 5 years. Now he is Associate Professor of School of Information and Communication Engineering at SKKU. Also, he has served as an executive committee member of Korean E-Government, and advisory committee members of several public and private organizations such as National Intelligence Service of Korea, Digital Investigation Advisory Committee of Supreme Prosecutors' Office, Ministry of Justice, The Bank of Korea, ETRI(Electronic and Telecommunication Research Institute), and KISA(Korea Information Security Agency), etc. His research interests include cryptography, information security and information assurance.